

KYMENLAAKSON AMMATTIKORKEAKOULU  
Energiatekniikka / Automaatio- ja prosessitekniikka

Axel Ahlqvist

MATEMAATTISEN MALLINTAMISEN JA SIMULOINNIN HYÖDYNTÄMINEN  
ENERGIA-ALALLA

Opinnäytetyö 2014

## TIIVISTELMÄ

KYMENLAAKSON AMMATTIKORKEAKOULU

Energiatekniikka

AHLQVIST, AXEL

Matemaattisen mallintamisen ja simuloinnin hyödyntäminen energia-alalla

88 sivua + 4 liitesivua

Työn ohjaaja

Yliopettaja Merja Mäkelä

Toimeksiantaja

Kymenlaakson ammattikorkeakoulu

Toukokuu 2014

Avainsanat

simulointi, prosessidynamiikka, dynaamiset mallit, siirtofunktiomallit, aikasarjamallit

Matemaattisella mallintamisella tarkoitetaan ilmiön tai prosessin tutkimista ja sen muuttamista matemaattiseen muotoon. Simuloinnilla pyritään tietokoneen avulla jäljittelemään prosessin tai ilmiön käyttäytymistä matemaattisten yhtälöiden avulla.

Nykyisen tietotekniikan kehittyminen on mahdollistanut erilaisten asioiden ja ilmiöiden mallintamisen sekä simuloinnin. Nykyisin saadaan pienellä investoinnilla riittävän tehokas tietokone ja matemaattinen ohjelmisto, jolla voidaan simuloida prosessin toimintaa. Opinnäytetyö keskittyy ensisijaisesti teollisten prosessien dynaamiseen mallintamiseen.

Mallinnusta tehdään periaatteessa kahdella päätävällä. Ensimmäinen tapa on tehdä matemaattinen malli fysikaalisten lainalaisuuksien avulla. Toinen tapa on kerätä tietoa prosessista ja sovittaa kerätty tieto matemaattiseen malliin. Matemaattisessa mallintamisessa yhdistyvät sekä matematiikka että käytäntö. Matemaattisesti luodaan yhtälö, joka ottaa huomioon fysikaaliset suhteet sekä muut muuttujat. Käytännössä hienosäädetään tärkeät muuttujat ja viritysparametrit.

Työssä kehitettiin käytännön mallinnustapoja ja koostettiin opetusmateriaali. Työn tuloksena voitiin todeta, että matemaattinen mallintaminen ja simulointi ovat hyödyllisiä ja käyttökelpoisia työkaluja prosessin käyttäytymisen selvittämiseen.

## ABSTRACT

KYMENLAAKSON AMMATTIKORKEAKOULU

University of Applied Sciences

Energy Technology

AHLQVIST, AXEL

Utilization of Mathematical Models and Simulation in  
Energy Industry

Bachelor's Thesis

88 pages + 4 pages of appendices

Supervisor

Merja Mäkelä, Principal Lecturer

Commissioned by

Kymenlaakso University of Applied Sciences

May 2014

Keywords

mathematical modeling, simulation dynamic models, discrete models

Mathematical modeling means converting phenomena, problems or processes to mathematical formulas. Simulation is to try imitate or repeat phenomena, problems or processes with computer.

Modern technology and information technology has enabled mathematical modeling and simulation by using computer. Today, it is possible to buy a powerful computer and mathematical software whereby is possible made simulation without huge investments. This thesis focuses to dynamic modeling of industrial processes.

Modelling has been possible by using two main methods. The first method is to make a mathematical model which has based on a physical law. The second method is to collect data from a process and adjust it to mathematical model structure. Mathematical modeling connects mathematics, physics and practical working. By using mathematics, equation must be take account of physical relationships and other variables. Practically, fine tuning is made for parameters and other variables that are important process parameters.

In this study, practical modeling methods has been developed, and thesis complied to teaching material. In this thesis, becomes clear that mathematical modeling and simulation are useful and workable application tools for investigating process dynamics.

# SISÄLLYS

## TIIVISTELMÄ

## ABSTRACT

1	JOHDANTO	6
2	MALLINTAMISEEN JA SIMULOINTIIN TARVITTAVA VARUSTUS	7
2.1	Tietokoneet	7
2.2	Matemaattiset ohjelmistot	9
2.2.1	Matlab ja Simulink	10
2.2.2	Scilab	13
2.2.3	Mathcad	13
2.2.4	SMath Studio	15
2.2.5	Teollisuuden automaatiojärjestelmien sovellusympäristöt	15
3	DYNAAMISTEN ILMIÖIDEN MALLINTAMINEN	16
3.1	Mallintamisen lähtökohtia	19
3.2	Yksi- ja monimuuttujamallit	20
3.3	Mallinnuksen vaiheet	21
4	MATEMAATTISIA MALLIRAKENTEITA PROSESSIDYNAMIIKALLE	25
4.1	Differentiaaliyhtälömallit	25
4.2	Laplace-tason mallit	28
4.3	Differenssiyhtälömallit	32
4.4	Z-tason mallit	35
4.5	Lohkokaavioesitys	35
4.6	Viiveiden käsittely	36
4.7	Prosessikokeet ja sopivan mallin valitseminen	38
6	SÄÄTIMIEN MALLINTAMINEN	39
6.1	PID-säätimien toiminta ja algoritmit	40
6.3	Säätimen viritys	43

7 DYNAAAMISTEN PROSESSIEN MALLINTAMINEN JA SIMULOINTI	45
7.1 Teoreettinen mallintaminen luonnonlakeja hyödyntäen	45
7.1.1 Säiliökapasiteetti integraattorimallina	46
7.1.2 Läpivirtaussäiliön pintasäätö	51
7.1.3 Läpivirtaussäiliön lämpökapasiteetti	56
7.2 Kerättyyn dataan perustuvat mallit	63
7.2.1 Painesäiliön ARX-malli	64
7.2.3 Painesäiliö	75
7.2.4 Prosessin tiedonkeruussa tapahtuvat ongelmat	80
7.3 Luonnonlakimalli vai kokeellinen malli?	82
8 YHTEENVETO	83
LÄHTEET	84
LIITTEET	

Liite 1. Ensimmäisen kertaluvun prosesseja

Liite 2. Toisen kertaluvun prosesseja

## 1 JOHDANTO

Matemaattisella mallintamisella tarkoitetaan käytännön ongelmien, ilmiön tai asian muuttamista matemaattiseen muotoon (Jäppinen, Kupiainen & Räsänen 2005, 64). Mallintaminen on tehtävä, joka vaatii luovuutta ja ongelmanratkaisukykyä (Marlin 1995, 54). Simuloinnilla tarkoitetaan asian tai ilmiön toiminnan jäljittelyä tietokoneen avustuksella ja sen esitystuloksina tutkitaan useimmiten trendikäyriä (Simulointi s.a). Matemaattinen mallintaminen käsittää laajan kirjon erilaisia ongelmia, joten tässä opinnäytetyössä keskitytään teollisuudessa esiintyviin ongelmiin ja niiden matemaattiseen mallintamiseen. Työn tavoitteena on selvittää, kuinka energia-alalla esiintyvien prosessien mallintaminen ja simulointi onnistuu käytännössä. Näkökulmana on soveltaa mallinnuksen teoriaa käytäntöön. Lisätarkoituksena on koostaa helppolukuinen teoriapaketti tuleviin Kymenlaakson ammattikorkeakoulun kehittyneiden säätömenetelmien kurssiin ja mahdollisesti muihin aihealuetta käsitteleviin kursseihin.

On olemassa kahdenlaisia malleja: staattisia ja dynaamisia malleja. Staattiset mallit ovat ajasta riippumattomia, kun taas dynaaminen malli ajasta riippuva. (Lyytikäinen & Mäkelä 2013.) Tässä opinnäytetyössä keskitytään dynaamisiin malleihin. Asioiden tai ilmiöiden mallintaminen on ollut tärkeää jo pitkän aikaa. Asioiden tai ilmiöiden mallintamisella tarkoitetaan kyseisten kohteiden tutkimista ja saadun tiedon muuttamista matemaattiseen muotoon käyttäen esimerkiksi differentiaaliyhtälöitä. Saatua matemaattinen muoto voidaan kirjoittaa tietokoneella olevaan matemaattiseen ohjelmistoon ja simuloida systeemin tapahtumia. Työelämässä malleja tarvitaan prosessin hallinnan tutkimista ja kehittämistä varten. Voimme mallintaa esimerkiksi lämmönsiirtymistä, kaukolämpöveden virtausta ja säätökäyttäytymistä. Simuloituamme prosesseja voimme löytää prosessista asioita, joita ei ole aiemmin huomattu. Tämä voi olla tärkeää, kun mallinnetaan tekeillä olevaa prosessia. Simuloinnissa suurin etu on se, että voimme kokeilla ja tehdä muutoksia ilman, että se vaikuttaa prosessiin. Kalliita prosessikohteita ei tällöin tarvitse tehdä.

Tässä opinnäytetyössä esimerkkeinä ovat käytännönläheiset energia-automaation sovellukset. Käytännössä mallintaminen ja simulointi vaativat kuitenkin matematiikasta tietoa integroinnista, derivoinnista, differentiaaliyhtälöistä, Laplace-muunnoksista, differenssiyhtälöistä ja Z-tason muunnoksista. Tämän opinnäytetyön tarkoituksena on selvittää kyseisiä matemaattisia laskutoimituksia.

## 2 MALLINTAMISEEN JA SIMULOINTIIN TARVITTAVA VARUSTUS

Simuloinnissa tietokone toimii apuvälineenä, jonka avulla saamme tiedot laskuista nopeammin kuin paperille laskemalla. Voimme kuitenkin jo hieman ennakoida tuloksia tutustumalla yhtälöön tarkemmin. Esimerkiksi löytämällä differentiaaliyhtälöstä sini-komponentin voidaan epäillä, että prosessissa esiintyy värähtelyjä. Sen sijaan  $e^n$ -komponentti kertoo, että prosessissa on eksponentiaalista kasvua.

Mallintamiseen ja simulointiin tarvittava varustus riippuu siitä, kuinka monimutkainen malli on kyseessä ja, kuinka paljon laskentatehoa se vaatii. Esimerkiksi teoreettisessa mallintamisessa voidaan aloittaa paperilla ja tietokoneella määrittelemään perusyhtälöitä, mutta kokeellisessa mallintamisessa joudutaan tekemään kokeita itse prosessin kanssa. Yleensä kuitenkin matemaattiset mallit pyritään pitämään mahdollisimman yksinkertaisina.

Simulointiosuus jakautuu kahteen haaraan: teollisuudessa esimerkiksi virtauksen simuloimiseen riittää tehokas teollisuustietokone, mutta tieteenalalla vaaditaan jo huomattavasti enemmän tehoa suuremman tietomäärän käsittelyä varten. Näitä suurempia tietomääriä käsitteleviä tietokoneita kutsutaan supertietokoneiksi. Lisäksi tulevaisuudessa voidaan tehdä simulointeja kvanttietokoneiden avulla, jolloin simulointi on nopeampaa kuin supertietokoneella.

### 2.1 Tietokoneet

Teollisuuden mallintamisen tarpeisiin riittää tehokas teollisuustietokone sekä matemaattinen ohjelmisto ja tarvittavat testiohjelmit. Nykyiset tietokoneet kehittyvät niin nopeasti, että tietokoneet hidastuvat noin 3 - 5 vuodessa, mutta esimerkiksi keiltauani matemaattisia ohjelmistoja, ne pyörivät vanhemmassa ja hitaammassa koneessa todella hyvin, eikä tietokoneen käyttö hidastunut yhtään. Tietotekniikka ottaa koko ajan valtavia harppauksia eteenpäin ja näin ollen kuluttajille on tarjolla yhä tehokkaampia ja monipuolisia tietokoneita. Lisäksi on myös mahdollista ostaa tietokoneen komponentit erikseen ja näin rakentaa mieluisan komponenttikokonaisuuden sisältävän tietokoneen.

Esimerkiksi Matlab-ohjelmisto vaatii vähintään seuraavat ominaisuudet:

- Tarvitaan Intel tai AMD x86 -suoritin.
- Kiintolevyllä pitää olla vähintään 1 Gb vapaata tilaa kiintolevyltä, mutta yleensä 3 - 4 Gb.
- Ohjelmisto tarvitsee vähintään 1024 Mb RAM muistia, mutta suositeltava määrä olisi 2048 Mb.
- Matlab toimii viimeisimmillä käyttöjärjestelmillä sekä uudemmilla mm. Windows 8, Windows 7, Windows Vista ja Windows XP. (System Requirements s.a)

Supertietokoneet ovat tiedealalla hyvin paljon käytössä. Niillä suoritetaan vaikeita laskelmia, simulointeja ja tiedonkäsittelyä, joissa normaalilla tietokoneella menisi todella pitkä aika ratkaista. Kirjoitushetkellä supertietokoneista tehokkain sijaitsee Kiinassa Oak Ridgen tutkimuslaitoksessa. Sen laskentateho on 30 petaflopsia. Yksi petaflops vastaa  $10^{15}$  laskutoimitusta sekunnissa (Jacobsen 2013, 21). Supertietokoneiden huonona puolena on niiden energiankulutuksen suuruus. Esimerkiksi Yhdysvalloissa jouduttiin sulkemaan vuonna 2008 avattu 100 miljoonaa dollaria maksanut Roadrunner supertietokone, koska sen ylläpitäminen tuli liian kalliiksi. Roadrunnerin vaatima teho oli täydellä laskentateholla 2345 kW. Uudet tehoiltaan vastaavat kuluttavat arviolta noin 500 kW. (Kotilainen 2013.)

Esimerkkeinä supertietokoneiden käytöstä voisin mainita sään ennustamisen. Säämallit perustuvat monimutkaisiin differentiaaliyhtälöihin. Nämä vaativat todella paljon laskentatehoa, joten supertietokone on pakollinen. Säämallissa ilmakehä esitetään kolmiulotteisena verkkona, jonka jokaiselle solulle määritetään perusmuuttujat, jotka ovat lämpötila, kosteus, ilmanpaine ja tuulen suunta ja nopeus. Kun tietokone on laskenut ilmakehän tilan kaikissa soluissa, se vertaa soluja keskenään. Tämän jälkeen tietokone hyppää ajassa eteenpäin kaksi minuuttia kerrallaan ja suorittaa laskelmat uudelleen. 30:stä säämallin laskemasta tilannekuvasta syntyy kuva ilmamassojen ja lämpöenergian liikkeistä ilmakehässä noin tunnin kuluessa. 2-3 päivän sää saadaan laskeksi vain kaikkein tehokkaimmilla supertietokoneilla. Nämä tehokkaimmat supertietokoneet laskevat 100 000 triljoonaa laskutoimitusta noin muutamassa tunnissa. (Hansen 2013, 66.)



Kvanttitietokone on ollut pitkän aikaa kehityksessä ja on epäilty sen onnistumista. Tutkija Scott Aaronson järjesti jopa kilpailun, jonka palkintona oli 100 000 euroa sille, kuka todistaa, että kvanttitietokone on mahdotonta rakentaa. Kvanttitietokone on vihdoin tullut todeksi. Kanadalainen D-wave-niminen yritys on kehittänyt kvanttitietokoneen. (Kotilainen 2013.) Kvanttitietokoneen tehokkuus perustuu siihen, että siinä hyödynnetään hiukkasluokan kvanttimaailman ilmiöitä, jolloin kvanttitietokone pystyy myös tulkitsemaan välitilat 10 ja 01, kun taas perinteisessä tietokoneessa voi olla vain tilat 0 ja 1 (Glanville 2013, 77). Vaikka kvanttitietokone onkin herättänyt valtavaa kiinnostusta, rajoittaa sen hankintaa 12 miljoonan euron hinta. Lisäksi nykyiset kvanttitietokoneet ovat vielä osittain rajoittuneita, eikä kaikkia niiden mahdollisuuksia pystytä vielä hyödyntämään. Kuitenkin nykyinen kehitys viittaa siihen, että kvanttitietokone on tulevaisuudessa mahdollinen saada toimimaan täysillä ominaisuuksilla. (Kotilainen 2013.)

## 2.2 Matemaattiset ohjelmistot

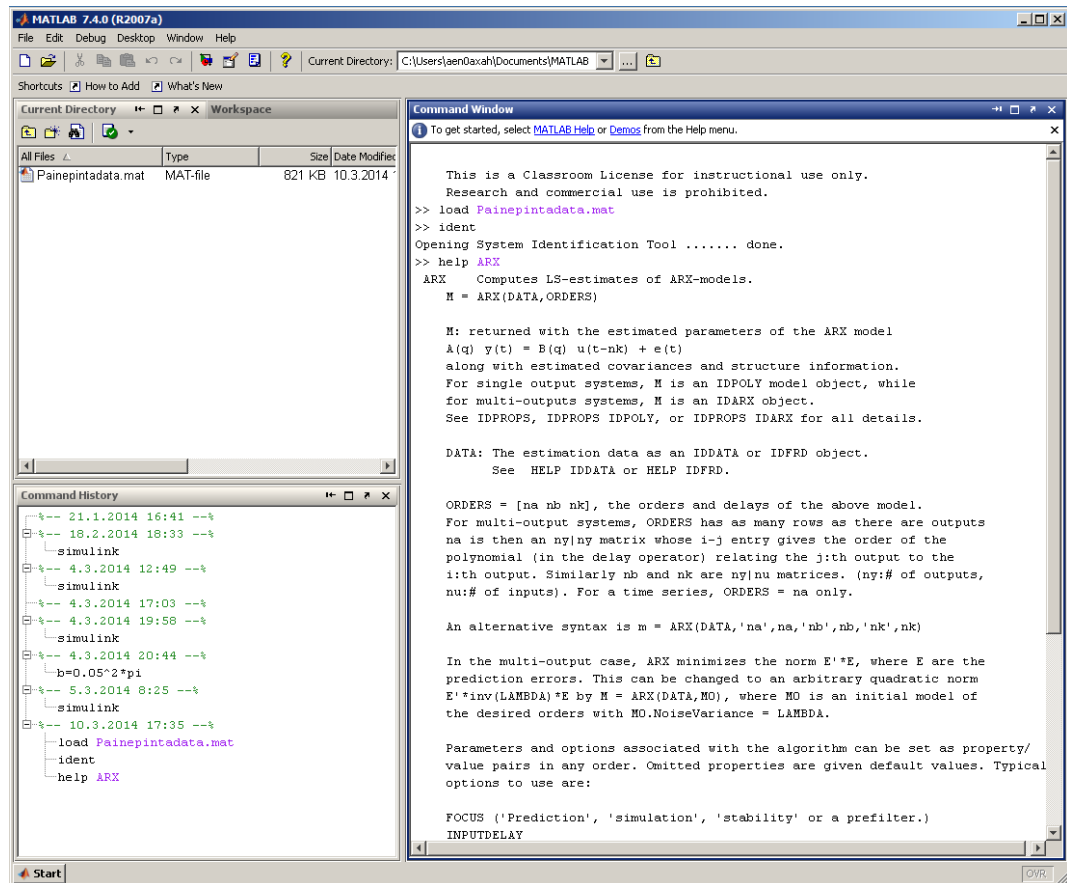
Mallintamisessa kannattaa käyttää matemaattista ohjelmistoa, koska ne nopeuttavat työtä jonkin verran. Tulokset tulisi varmistaa ja myös se, että kaikki syötetyt arvot ja alkulaskut ovat oikein. Lisäksi kannattaa käyttää varsinkin alkuvaiheessa standardeja SI-yksiköitä, koska tulokset voivat heitellä hyvin paljon muilla arvoilla. Esimerkiksi NASAn vuonna 1998 laukaisema Mars Climate Orbiter -luotain, joka katosi ennen kuin se ehti asettua Mars-planeetan kiertoradalle. Kaavat olivat oikein, mutta jotkut tutkijoista olivat käyttäneet yksikköinä tuumia ja nauvoja, toisilla tutkijoilla oli käytössä metrijärjestelmä. (Glanville 2013, 103.)

Vaikka matemaattiset ohjelmistot helpottavat huomattavasti laskutoimituksissa, täytyy muistaa kuitenkin, että aina käytössä ei ole tietokonetta tai ohjelmistoja joten on siis aina hyvä muistaa myös ylläpitää matemaattisia taitoja, kuten lausekkeiden sieventämistä. Lisäksi myös simuloinnin jälkeiset tulokset kannattaa analysoida huolella mahdollisten virheiden varalta, sillä tulosten analysoimatta jättäminen voi vaikuttaa tulokseen ja siihen, että esimerkiksi ohjelmiston koodissa mahdollisesti olleen virheen huomaamisen todennäköisyys pienenee.

### 2.2.1 Matlab ja Simulink

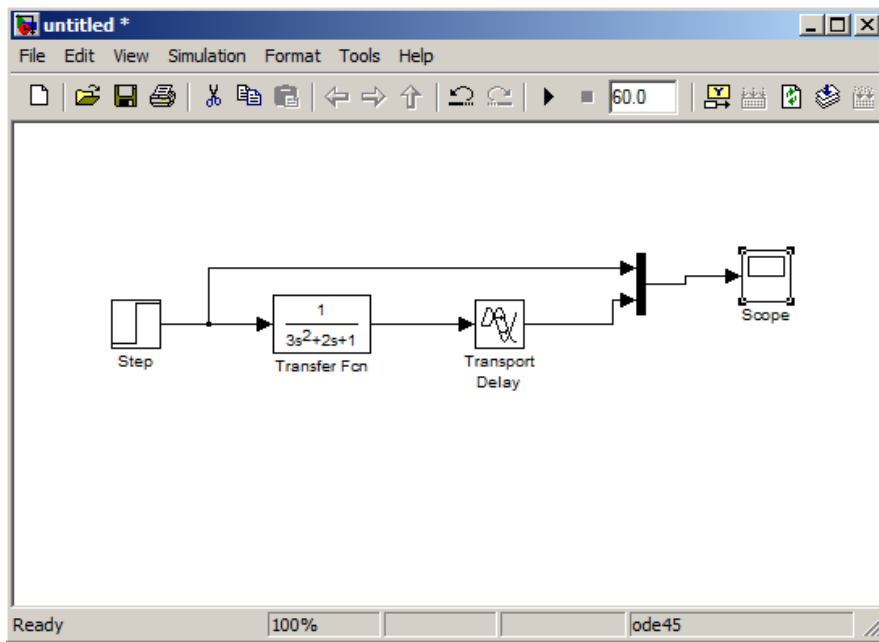
Matlab on yhdysvaltalaisen MathWorks -yhtiön kehittämä matemaattinen ohjelmisto, jolla on käyttäjiä maailmanlaajuisesti yli miljoona. Se on suunniteltu monipuoliseksi ohjelmistoksi, jolla pystyy esimerkiksi ohjelmoimaan algoritmeja, tiedon analysointia, visualisointia sekä numeerista laskentaa, mutta se vaatii jonkin verran harjoittelua, ennen kuin sitä oppii hyödyntämään tehokkaasti. Matlab-ohjelmistoon liittyvää kirjallisuutta löytyy todella paljon, tilastojen mukaan 1500 kirjaa, jotka ovat 28 kielelle. (Company Overview s.a.) Lisäksi esimerkiksi Matlabin kotisivuilta ja Youtube.com-sivustolta löytyy runsaasti videoita, joissa opetetaan ohjelmiston käyttöä ja toimintojen suorittamista. Alkuun pääsee hyvin Modern Control Systems -kirjan liitteen 1 avulla. (Dorf, Bishop 2011, 1060 - 1076).

Yksi Matlabin pääpiirteistä on sen vapaa muokattavuus. Käyttäjä voi muokata vapaasti ohjelmiston valmiita funktioita, mutta samalla luoda omia funktioita. Koodin muokauksessa on ongelmana, että ohjelmiston käyttäjä voi poistaa mahdollisesti tärkeän ominaisuuden. Tämä huomattiin Matlabia käytettäessä, kun yritimme luoda funktiolle kuvaajaa ja ihmettelimme, miksi emme saa luotua helpointa mahdollista kuvaajaa plot-komennolla. Tutkittuamme asiaa huomasimme, että kyseinen komento oli tallella, mutta koodi oli poistettu. Näin ollen ohjelmisto luuli, että kyseinen funktio oli vielä ohjelmistossa, mutta ei voinut suorittaa sitä, koska siinä ei ollut koodattuja komentoja, joiden avulla ohjelmisto olisi luonut kuvaajan.



Kuva 1. Matlab-ohjelman käyttöliittymä

Simulink on Matlabiin kuuluva lisäosa, jota voidaan käyttää moniulotteisten dynaamisten sekä upotettujen systeemien mallien simulointiin. Simulinkin toiminta perustuu lohkokaaaviomaiseen signaalin käsittelyyn. (Simulink Product Description s.a). Simulinkillä voidaan analysoida systeemin toimintaa ja muokata esimerkiksi pitkästä differentiaaliyhtälöstä koostuva siirtofunktio pienempään muotoon. Simulinkin avulla tarkastellaan systeemin vastetta trendikäyrien avulla. Useimmiten analysoitavat trendikäyrät tulkitaan kaksikulotteisina, mutta tarvittaessa Matlabilla voidaan näyttää trendikäyrät myös kolmiulotteisina. Tämä vaatii kuitenkin, että meillä on arvot  $x$ ,  $y$  ja  $z$ -akseleille. Kaksikulotteisessa trendikäyrässä  $x$ -akseli on yleensä aika ja  $y$ -akseli on suure tai ohjausarvo. Opinnäytetyössäni hyödynnän kyseistä lisäosaa.



Kuva 2. Simulink-ohjelman käyttöliittymä

Matlabiin on saatavilla useita erilaisia maksullisia työkaluja, joita nimitetään nimellä toolbox. Toolboxit sisältävät valmiiksi ohjelmoituja koodeja sekä algoritmeja sille osoitettuun tarkoitukseen. Kymenlaakson ammattikorkeakoulun käyttämä Matlab-ohjelma sisältää ainakin seuraavat toolboxit:

- Symbolic Math Toolbox
- Signal Processing Toolbox
- System Identification Toolbox
- Fuzzy Logic Toolbox
- Filter Design Toolbox
- Fixed-point Toolbox
- Control System Toolbox.

Matlabin lisenssi julkiseen käyttöön maksaa 2000 € ja yksityisessä kotikäytössä kyseisen ohjelman lisenssi kustantaa 129 €. Kotikäyttöön tarkoitettussa Matlab-ohjelmistossa tulee huomioida, että se ei sisällä Simulink-ohjelmaa, toisinkuin julkiseen käyttöön tarkoitettu versio. Simulink-ohjelma kotikäyttöversioon kustantaa 29 €. Tuotteiden hinnoissa ei oteta huomioon arvonlisäveroa. (Products for MATLAB & Simulink Version R2014a s.a), (Products for MATLAB Home R2014a. s.a)

### 2.2.2 Scilab

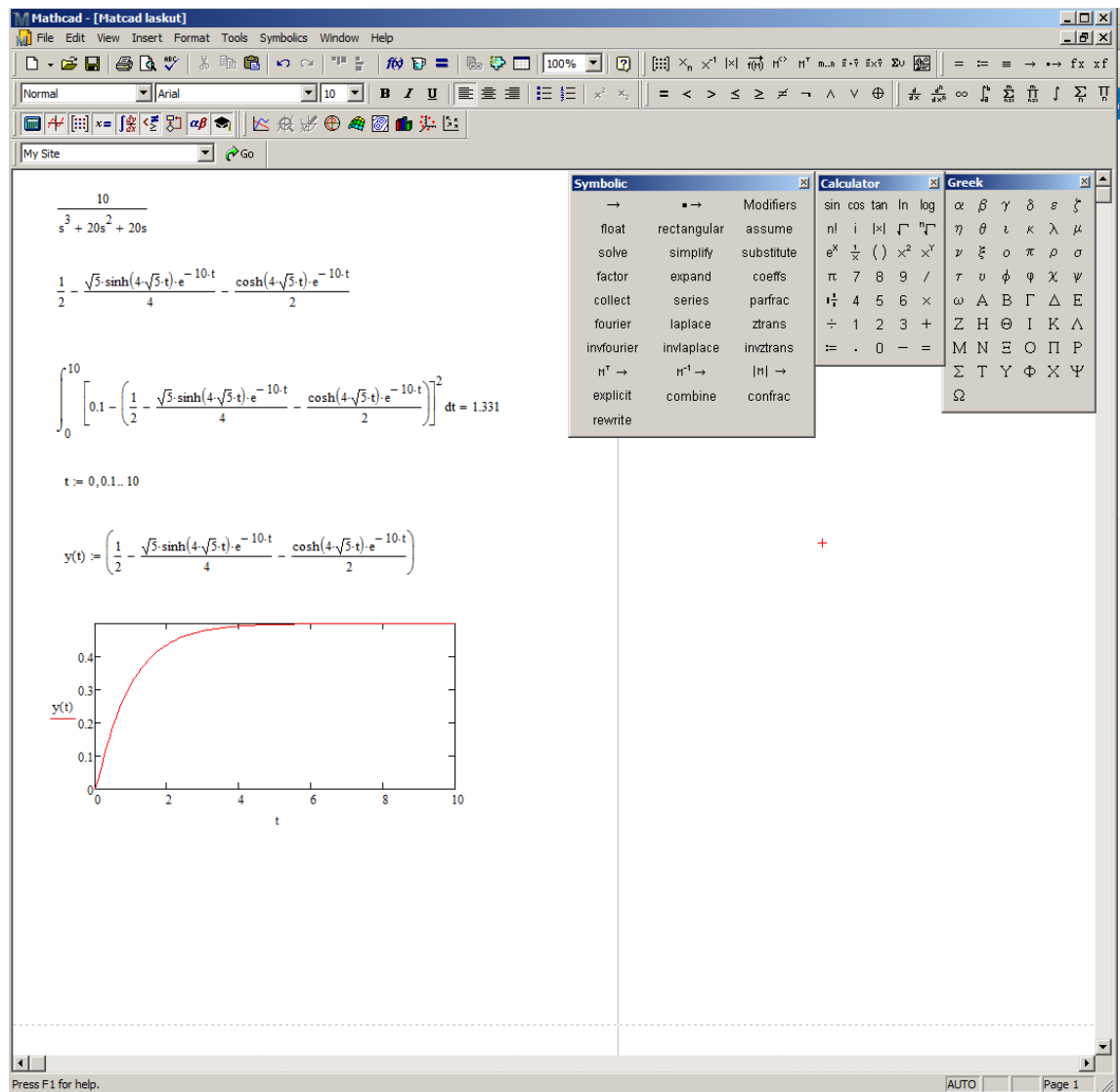
Scilab on ilmainen vastine Matlab-ohjelmistolle. Kyseessä on ranskalaisen Scilab Enterprises nimisen yhtiön kehittämä ilmainen ohjelmisto matemaattiseen ongelmien ratkaisuun. Ohjelma muistuttaa käyttöliittymältään hyvin paljon Matlabia, eli siinä on tekstipohjainen komennon syöttäminen. Scilab on ilmainen ohjelmisto kaikille ja sen voi ladata ohjelman kotisivuilta: <http://www.scilab.org/>. Scilabin kaupallinen käyttö on ilmaista, mutta kaupallisessa käytössä oleva ohjelma ei saa olla muokattu tai johdettualkuperäisestä ohjelmasta ilman valmistajan lupaa (SCILAB Licence 2014.)

Scilab Enterprises on perustettu vuonna 2010 ja se on kehittänyt avoimen lähdekoodin matemaattisen ohjelmiston Scilabin. Scilab Enterprises yhtiön henkilökunta koostuu mm: insinööreistä ja tohtoreista. Scilabia käytetään mm: ilmailualalla, autoalalla, energia-alalla, puolustusvoimissa, finanssi- sekä kuljetusalalla. (About Scilab Enterprises s.a.) Lisäksi Scilab Enterprises on kehittänyt vastaavanlaisen ohjelman, joka vastaa Matlabissa olevaa Simulink-ohjelmaa. Scilabin kehittämän ohjelman nimi on Xcos.

Kokeilujeni perusteella Scilab vaikuttaa vahvalta kilpailijalta Matlabille. Ohjelman ainoa heikkous on, että sen käyttö on haastavaa, koska se ei käytöltään vastaa täysin Matlabia. Tämä tarkoittaa sitä, että ohjelman käyttö on vaikeaa suoraan Matlabista Scilabiin siirtyvälle, mutta kuitenkin ohjelmiston käyttämiseen löytyy paljon oppaita ja ohjevideoita internetistä.

### 2.2.3 Mathcad

Mathcad on PTC-yhtiön kehittämä matemaattinen ohjelmisto. Ohjelma on käyttöliittymältään erilainen verrattuna Matlabiin ja Scilabiin. Mathcadin käyttöliittymässä valitaan matemaattiset toiminnot oikealta tai ylhäällä olevasta valikosta. Mathcad ei sisällä samanlaista dynaamisten mallien simulointiin tarkoitettua työkalua, kuin Matlab tai Scilab, mutta sillä on kätevä suorittaa esimerkiksi vaikeita laskutoimituksia. Mathcadilla pystyy esimerkiksi suorittamaan mm: Laplace-muunnoksen differentiaaliyhtälöille ja päinvastoin. Mathcadin vahvuus on sen helppokäyttöisyys. Sen avulla voidaan helposti laskea esimerkiksi fysiikkaan liittyviä ongelmia ja kokeilla ilmiöiden muuttumista erilaisilla arvoilla, esimerkiksi suorituskykyindeksin laskeminen, joka esiintyy kuvassa 3.



Kuva 3. Mathcad -ohjelmiston käyttöliittymä

Mathcad -ohjelmiston voi ostaa suoraan maahantuojalta ja oston voi suorittaa suoraan sivustolta [www.mathcad.fi](http://www.mathcad.fi). Mathcadin normaaliversio maksaa 1540 € ja opiskelija versio maksaa 40 €. Normaaliversion hinta ei sisällä arvonlisäveroa, kun taas opiskelijaversion hinta sisältää arvonlisäveron. (Tilaa Mathcad s.a), (Mathcad opiskelijaversio s.a).

#### 2.2.4 SMath Studio

SMath on internetistä ilmaiseksi ladattava matemaattinen ohjelmisto. Kyseessä on matemaattinen ohjelmisto, jonka tarkoituksena on olla alusta matemaattisille laskutoimituksille ikään kuin piirtäisit paperille laskutoimituksia. Käyttöliittymältään SMath studio on lähestulkoon samankaltainen, kuin Mathcad 15, ja ohjelmisto vie hyvin vähän muistia tietokoneelta, joten ohjelmaa voi käyttää hitaammallakin koneella.

SMath-ohjelmiston ongelmana on, että se on toiminnoiltaan hieman vajaa verrattuna esimerkiksi Matlabiin. Yritin esimerkiksi kuvassa 3 esiintyvää laskua avata SMath-ohjelmalla, mutta ohjelma antoi minulle virheilmoituksia liittyen laskujen osuuksiin. Esimerkiksi fysiikan laskuja suoritettaessa ohjelma on hyvin nopea ja helppokäyttöinen. SMath ohjelman voi ladata ilmaiseksi sen kotisivuilta. SMathin hinnasta kaupalliseen käyttöön ei löytynyt tietoa.

#### 2.2.5 Teollisuuden automaatiojärjestelmien sovellusympäristöt

Teollisuuden automaatiojärjestelmissä automaatio suunnittelu tehdään usein valmistajan toimittamalla omalla työkalulla. Esimerkiksi Metso DNA:ssa voidaan säätöventtiilin toiminta määrittellä FbCAD-nimisen ohjelmiston avulla. Toisaalta käytössä voi olla myös esimerkiksi Honeywell Experion PKS.

FbCAD-ohjelmalla luodaan graafinen lähdekoodi prosessin toiminnoista. Kyseisellä ohjelmalla voidaan määrittää esimerkiksi hälytysrajat, pakko-ohjaukset, yhteydet muihin piireihin jne. Metson Picture Designerilla voidaan luoda ajokaavio ohjattavasta prosessista ja yhdistää toiminnot suoraan FbCADilla luotuun piiriin. Luomamme piiri voidaan testata järjestelmässä suoraan, mutta useimmiten kuitenkin räätälöidään testiympäristö tulevan prosessin mukaan.

Simuloinnin tuloksena voidaan tehdä tulosten pohjalle simulaattori. Simulaattori poikkeaa simuloinnista siten, että simulaattori jäljittelee prosessin käyttäytymistä reaaliajassa. Esimerkiksi Matlabin Simulinkillä suoritettu simulointi kertoo 5 minuutin tapahtumat muutamassa sekunnissa, kun taas simulaattorissa pyrimme hallitsemaan prosessia reaaliajassa. Simulaattorin tekemisessä voidaan käyttää hyväksi simuloinnista saatua dataa, joka ennakoii, mitä prosessille tapahtuu muutoksen jälkeen.

Teollisuuden sovellusympäristöissä simulaatiot prosessista ovat hyvin tärkeitä, koska näin operaattori pystyy harjoittelemaan esimerkiksi vaaratilanteita varten ennakolta, harjoittelemaan prosessin stabilointia, harjoittelemaan tehtaan ylösajoa jne. Kaikki tämä voidaan suorittaa turvallisesti testiympäristössä ilman, että itse prosessiin tulee häiriöitä tai muutokset prosessissa aiheuttaisivat tuotannollista haittaa tai vaaratilanteita. (Joronen, Kovács & Majanne 2007, 211 - 212.)

### 3 DYNAAMISTEN ILMIÖIDEN MALLINTAMINEN

Matemaattisessa mallintamisessa hyödynnetään matematiikkaa, fysiikkaa ja käytäntöä. Matemaattisilla malleilla pyrimme ymmärtämään ongelmia ja ilmiöitä ja samalla saamaan niille matemaattinen muoto. Muotoilemalla ongelma tai ilmiö matemaattiseen muotoon, se on huomattavasti helpompi ymmärtää. Lisäksi matematiikka on kaikkialla sama, ainoastaan merkinnät saattavat olla erilaisia, esimerkiksi amerikkalaisissa yliopistoissa saatetaan käyttää erilaisia merkintöjä, kuin Suomessa, mutta pohja on sama. Matemaattisesta mallista voimme nähdä prosessin käyttäytymisen, mutta pidemmän aikavälin tapahtumien hahmottamiseen tarvitaan simulointia, jossa pyrimme matemaattisen mallin kautta toistamaan sellaisena, kuin se on aikaisemmin esiintynyt. Simuloinnin tuloksena on trendikäyrä, josta useimmiten tutkimme systeemin käyttäytymistä verrattuna aikaan. Joskus kuitenkin on tilanteita, jolloin joudumme vertaamaan kahden eri suureen muuttumista, mikäli toinen suure muuttuu. Esimerkkinä kahden suureen muuttumisen vertaamisena voitaisiin mainita tuuliturbiinin teho, kun tuulen nopeus pienenee.

Matemaattisella mallintamisella halutaan tutkia erilaisia tasetilanteita ja, kuinka tasapainotilan muutos vaikuttaa tasetilanteeseen. Energia-alalla mallintamisesta on pääosin hyötyä tutkittaessa sitä, kuinka hyvin säädin pystyy korjaamaan prosessin tasapainotilan muutoksen. Vaikka tämä kuulostaa vähäiseltä, niin huonosti tehty säätö voi parhaimmillaan aiheuttaa paljon kustannuksia. Mallintamismielessä tutkitaan myös prosessin dynamiikkaa. Dynamiikalla tarkoitetaan voimien vaikutusta kappaleiden liikkeisiin. Alun perin dynamiikkakäsite koski vain kiinteitä kappaleita, mutta myöhemmin dynamiikkaan liitettiin mm: vesivirtaukset (Glanville 2013, 36 - 37). Prosessidynamiikalla kuitenkin tarkoitetaan tulon (merkitään Y-kirjamella, lähtö merkitään U-kirjaimella) muutoksen vaikutuksen etenemistä prosessissa ajan mukana (Harju & Marttinen 2000, 9). Kuitenkin prosessidynamiikasta meitä kiinnostavat erityisesti vii-



veet, prosessin aikavakio ja prosessin vahvistus. Näillä arvoilla on tärkeä rooli yksinkertaisia kokeellisia malleja luodessa. Prosessin dynaamisuus tarkoittaa sitä, että tulon muutokset eivät näy lähdön puolella välittömästi, vaan muutos näkyy vasta tietyn siirtymäajan jälkeen (Henttonen, Peltomäki & Uusitalo 2004, 144).

Kuten myöhemmin huomaamme, dynaamisten ilmiöiden mallintaminen jakautuu teollisuudessa kahteen päähaaraan: teoreettiseen mallintamiseen ja kokeelliseen mallintamiseen. Teoreettisessa mallintamisessa pohjaututaan hyvin paljolti fysikaaliseen teoriaan, kun taas kokeellinen mallintaminen pohjautuu kokeilulla saatuun tietoon. Yhteistä kummallekin tapaukselle on, että malli on likimääräinen kuvaus haluamastamme ilmiöstä. On hyvin vaikea sanoa, kumpi tapa mallintaa olisi parempi ja helpompi, sillä kummallakin tavalla mallintaminen vie aikaa ja sisältää omat vaikeutensa.

Dynaamisissa malleissa voidaan pyrkiä samaan tietoa muun muassa seuraavista asioista:

- systeemin tilan muutokseen kuluva aika
- systeemin stabiilisuus
- systeemi vastaus vasteeseen
- toisen suureen vaikutus systeemiin.

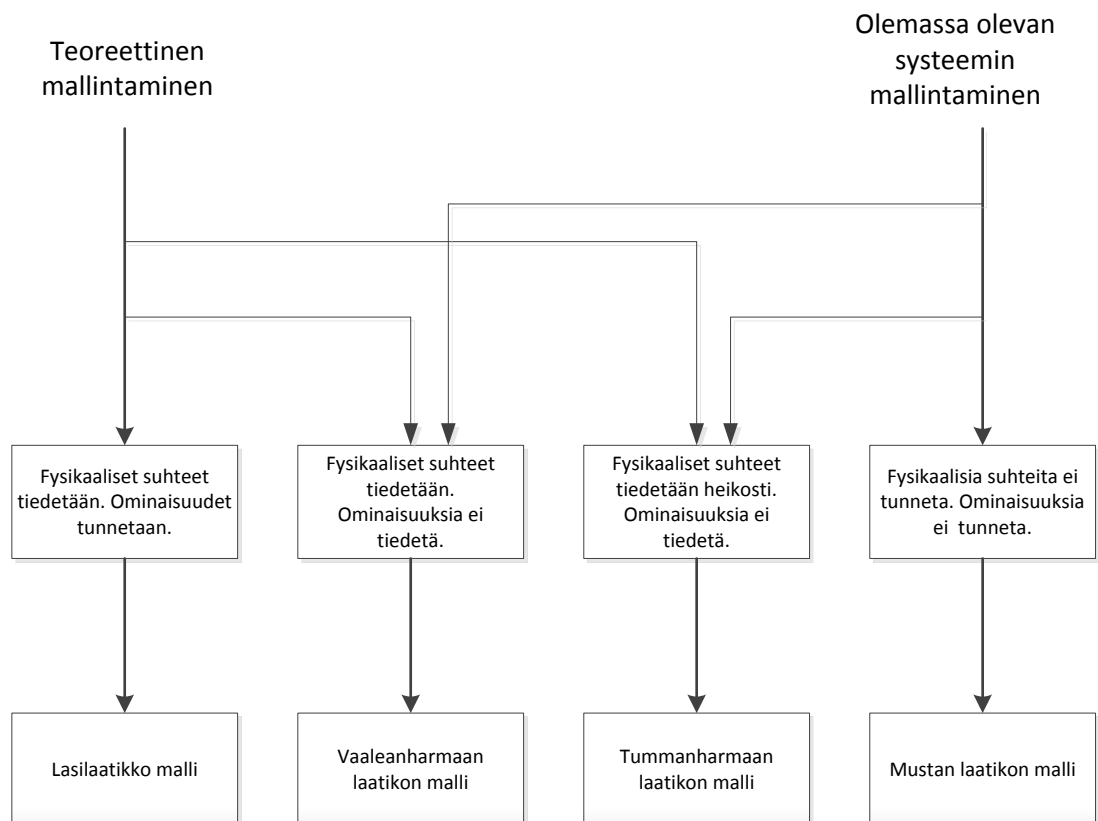
Matemaattinen malli on likimääräinen kuva prosessista. Siihen tulee mukaan tärkeimmät elementit, jotka vaikuttavat prosessiin. Sekä teoreettisessa että kokeellisessa mallintamisessa on omat virhemarginaalinsa, joita ei voi lukuarvollisesti määrittää. Teoreettisessa mallissa virhettä tuovat mm: teoreettinen kaava, likiarvojen käyttö, pyöristysvirheet. Kokeellisessa mallintamisessa suurimmat virheen aiheuttajat ovat esimerkiksi prosessin ohjauksen viiveet, tiedon näytteenotto tarkkuus, näytteenottoväli jne. Mallinnettaessa oletamme, että systeemi käyttäytyy lineaarisesti. Mikäli systeemi kuitenkin käyttäytyy epälineaarisesti, voimme kuitenkin approksimoida systeemin käyttäytymistä lineaarisesti esimerkiksi Taylorin sarjan avulla, mutta tällöin mallin käyttöalue on kapeampi.

On olemassa useita erilaisia systeemejä, joita me voimme tutkia. Teollisuudessa ja tekniikanalalla kiinnostavia prosesseja ovat

- sähköinen systeemi

- lämpösystemi
- virtaussysteemi
- mekaaninen systeemi.

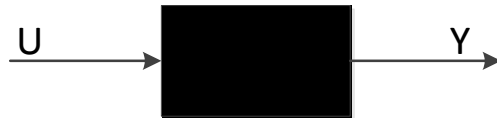
Yhteistä näille suureille on se, että niiden välillä on tietynlainen analogia eli ne muistuttavat toisiaan. Tämä tarkoittaa sitä, että voimme soveltaa kaavoja toiseen prosessiin ja huomata samanlaista käyttäytymistä. Kuvassa 4 on mallintamisen lähtökohtia. Kaavion avulla pystytään selvittämään, kuinka vaikea mallintamistapaus on kyseessä. Lasilaatikkomalli on helpoin ja mustan laatikon malli on haastavin ja mahdollisesti eniten aikaa vievin mallinnuksen ongelma. Mallinnettaessa oletamme, että systeemi käyttäytyy lineaarisesti. Mikäli systeemi kuitenkin käyttäytyy epälineaarisesti, voimme kuitenkin approksimoida systeemin käyttäytymistä lineaarisesti, mutta tällöin mallin käyttöalue on kapeampi.



Kuva 4. Erilaisten mallien lähtökohdat (Staben 2013.)

### 3.1 Mallintamisen lähtökohtia

Mustan laatikon malli on ongelma tilanne, jossa emme tiedä mitään systeemistä. Tässä tapauksessa ei ole tietoa fysikaalisista suhteista, eikä systeemin ominaisuuksista, mutta voimme kuitenkin saada tietoa prosessin tapahtumista mittaamalla sisääntulon ja ulostulon ja luomalla siirtofunktion niiden välisestä suhteesta. Mustan laatikon ongelman saa ratkaistuksi vain mittaamalla tulon ja lähdön ja luomalla tietojen pohjalta mallin. (Staben 2013.)



Kuva 5. Mustan laatikon malli

Tumman harmaassa mallissa on mallintamisen ongelmatilanne, jossa tiedämme heikosti fysikaalisista yhteyksistä, mutta meillä ei ole tietoa systeemin ominaisuuksista. Tumman harmaan laatikon malli voidaan ratkaista samalla tavalla, kuin mustan laatikon malli eli mittaamalla ulostulo ja sisääntulo ja luomalla malli. Tumman harmaan laatikon mallissa malliin voidaan yhdistää jo hieman tietoa fysikaalisista suhteista. (Staben 2013.)



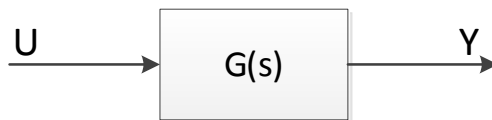
Kuva 6. Tummanharmaan laatikon malli

Kirkkaan harmaan laatikon mallissa tunnetaan fysikaaliset yhteydet, mutta ominaisuudet eivät ole tiedossa. Kirkkaan harmaan laatikon malli voidaan ratkaista mittaamalla tulo ja lähtö, kun tiedetään tulon ja lähdön arvot voidaan ne yhdistää melko helposti fysikaalisiin suhteisiin ja saadaan helposti tietoon kuvaus prosessin toiminnasta. (Staben 2013.)



Kuva 7. Tummanharmaan laatikon malli

Lasilaatikkomalli on helpoin mahdollinen tilanne systeemin mallinnuksessa. Tällöin meillä on tiedossa systeemin fysikaaliset suhteet ja ominaisuudet. Näiden tietojen pohjalta on helppo ymmärtää prosessin toimintaa. Pääosin teoreettisessa mallintamisessa päädytään lasilaatikkomalliin. (Staben 2013.)

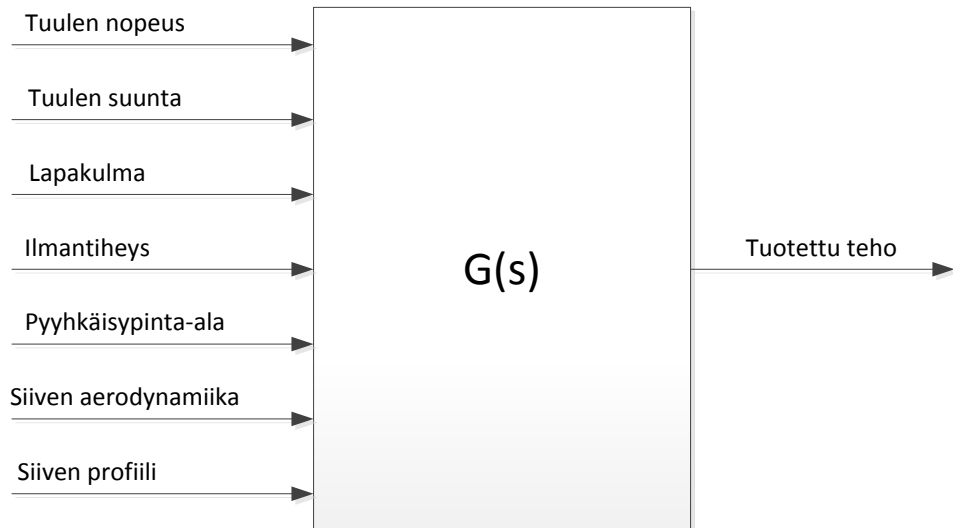


Kuva 8. Lasilaatikkomalli

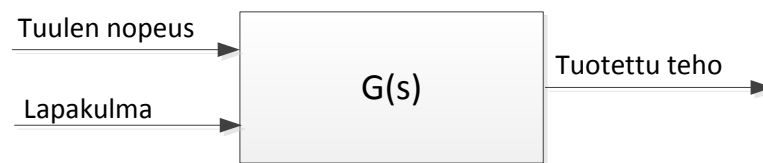
### 3.2 Yksi- ja monimuuttujamallit

Yksimuuttujamallissa meillä on yksi muuttuja, joka vaikuttaa systeemissä yhteen asiaan (Savolainen & Vaitinen 1995, 7). Tästä esimerkkinä voidaan mainita läpivirtaus-säiliö kappale 6.1.2, jossa tuloveden virtausmäärä vaikuttaa pinnankorkeuteen. Yksimuuttujamallit ovat helpoimpia mahdollisia malleja, ja niitä on helppo simuloida sekä analysoida.

Monimuuttujamallissa on tilanne, jossa useat muuttujat vaikuttavat toisistaan riippumatta moniin prosessisuureisiin. (Savolainen & Vaitinen 1995, 7) Hyvänä esimerkkinä tästä on säiliö, johon kaadetaan kuumaa ja kylmää vettä. Tiettyyn lämpötilaan päästäksemme meillä täytyy olla oikea suhde kuumaa ja kylmää vettä, mutta lisäksi täytyy huolehtia pinnankorkeudesta. Kyseinen säiliö esimerkki löytyy kappaleesta 6.1.3. Toinen esimerkki monimuuttujamallista on tuuliturbiini. Tuuliturbiinin tuottamaan tehoon vaikuttavat monet asiat, kuten kuvasta 9 voimme huomata. Aikaisemmin mainitsin, että matemaattiseen malliin otetaan vain oleelliset tiedot. Kuvassa 10 voimme nähdä yksinkertaistetun version tuuliturbiiniin vaikuttavista tekijöistä.



Kuva 9. Tuuliturbiinin lohkokaaavioesitys (Haapanen 2004)



Kuva 10. Yksinkertaistettu tuuliturbiinin lohkokaaavioesitys

Matemaattisessa mallintamisessa pyrimme yksinkertaistamaan kuvaa prosessista, jotta joudumme selvittämään asioiden yhteyksiä ja poistamaan kaavasta kohdat, jotka eivät vaikuta prosessiin tai vaikuttavat todella pienellä kertoimella. Mustan laatikon mallia varten meidän pitäisi tietää vähintään tuulen nopeus, lapakulma ja tuotettu teho. Näin saisimme selville, millainen mekanismi tuuliturbiinilla on.

### 3.3 Mallinnuksen vaiheet

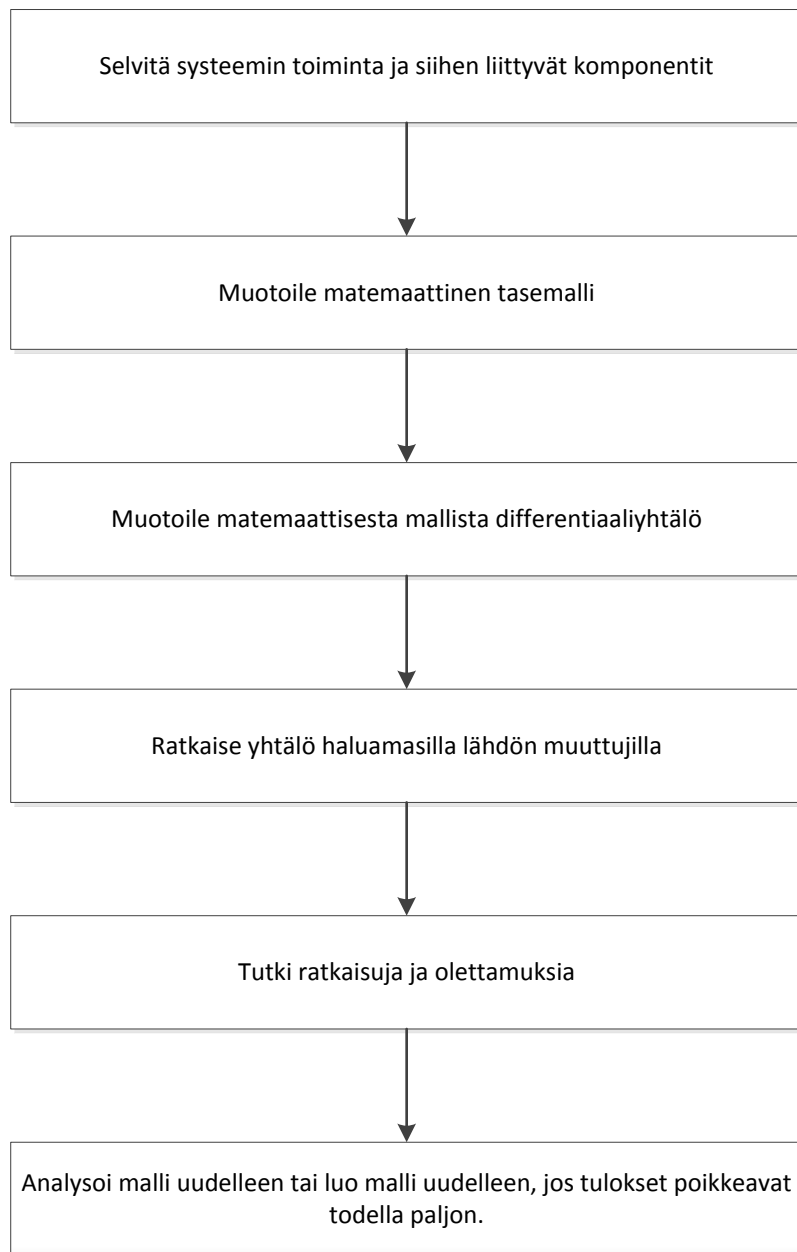
Prosessien ja säädön hallinnassa edellytyksenä ovat se, että fysikaalisten suureiden väliset riippuvuudet tunnetaan. Tämä tarkoittaa, että jo mallinnuksen suunnittelu vaiheessa täytyy tietää, mitä suureita aiotaan seurata tai, minkä suureiden välistä muutumista aiotaan seurata. Fysikaalisten suureiden riippuvuudet täytyy tuntea sekä tasapainotilassa että muutostilassa. Kuvassa 11 ja 12 on step-by-step -menetelmällä mallinnuksen vaiheet.

Matemaattista mallinnusta aloittaessa pitää kartoittaa, onko prosessi jo olemassa vai onko prosessi vasta tulossa. Tämä vaikuttaa siihen, mitä kautta mallia aloitetaan luomaan. Prosessin ollessa jo olemassa, suoritetaan prosessien mallintaminen kokeellisin menetelmin, kuten esimerkiksi askelkokeella. Mikäli prosessi on vasta suunnitteilla, suoritetaan mallintaminen teoreettisella mallintamisella. Kummassakaan tavassa ei varsinaisesti tule minkäänlaisia etuja, vaan kummassakin tavassa on omat hyvät ja huonot puolensa.

Kokeellinen mallintaminen aloitetaan prosessikokeen suunnittelusta. Tällöin selvitetään se, milloin ja, miten koe tehdään, koska prosessin ohjausarvoja ei voida muuttaa normaalissa ajotilanteessa. Tällöin voi vahingossa aiheutua hengenvaarallinen tilanne tai pahimmillaan jopa onnettomuus. Kokeiden suorituksen jälkeen ratkotaan matemaattisesti mahdolliset muuttujat ja ne sovitetaan malliin. Ja näin saadaan esimerkiksi mustanlaatikonmalli. Toinen mahdollisuus on, että sovitetaan parametreja, jolloin pääsemme yhtälön muokkaukseen ja sitä kautta jatkuva-aikaiseen malliin ja sen siirtofunktion. (Leiviskä 1999, 271.)

Teoreettisessa mallintamisessa aloitetaan tutkimalla tulevaa prosessia ja yritetään löytää muuttujat. Muuttujien tunnistamisen helpottamiseksi kannattaa tehdä laatikkomalli, kuten esimerkiksi kuvassa 10. Saatuamme erilaiset muuttujat selville luodaan analyttiset yhtälöt jotka tässä tapauksessa ovat todennäköisesti differentiaaliyhtälöitä. Tarpeen mukaan yhtälöt muokataan ja näin saadaan jatkuva-aikainen siirtofunktio. (Leiviskä 1999, 271.)

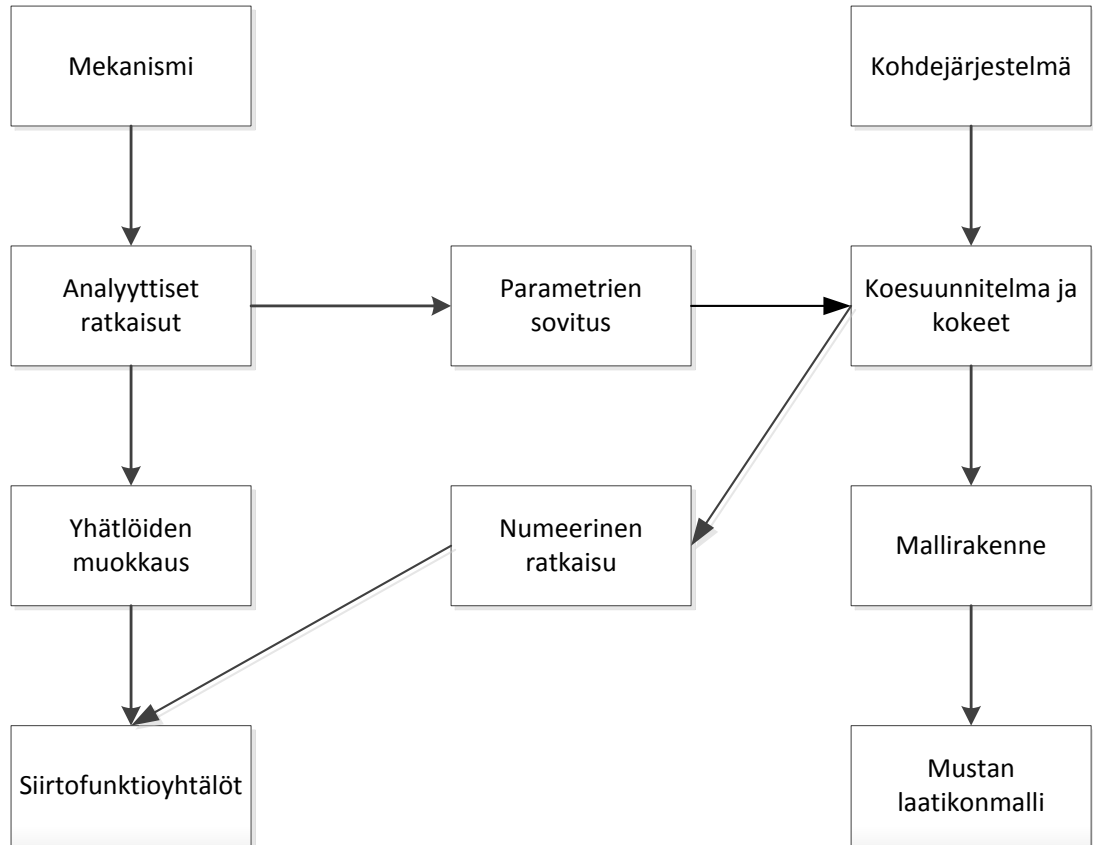
Selvitetyämme siirtofunktiot voimme ajaa prosessin esimerkiksi Matlab-ohjelmiston Simulink-ohjelmassa. Trendikäyrästä voimme tulkita, millaisen vasteen prosessi antaa. Lisäksi mallia kannattaa testata useilla eri suureilla, kokeilla raja-arvoja ja kuormittaa eri tavoin prosessia, jotta voimme nähdä, esiintyykö prosessissa ongelmia.



Kuva 11. Lohkokaavio esitys mallintamisesta (Dorf, Bishop 2011, 41)

## Teoreettinen mallinnus

## Kokeellinen mallinnus



Kuva 12. Mallintamisen vaiheet (Leiviskä 1999, 271)



#### 4 MATEMAATTISIA MALLIRAKENTEITA PROSESSIDYNAMIIKALLE

Matemaattiset mallirakenteet toimivat pohjana simulointia varten. Mallirakenne määräytyy sen mukaan, onko kyseessä teoreettinen malli vai kokeellinen prosessimalli. Molemmista tapauksista matemaattinen mallirakenne sisältää prosessin dynamiikan eli tiedon prosessin käyttäytymisestä. Teoreettisen ja kokeellisen mallintamisen

Teoreettisen mallin dynamiikka kehitetään fysiikan lakien avulla, jossa muuttujat ovat sidottuina aikaan. Tämä poikkeaa normaaleista fysiikan laskuista, joissa oletamme, että tilanteet pysyvät vakiona. Teoreettinen mallintaminen voidaan pääpiirteittäin erottaa kokeellisesta mallintamisesta siten, että yleensä teoreettisessa mallintamisessa käytetään differentiaaliyhtälöjä.

Kokeellisissa malleissa saamamme mittausdata sovitetaan valmiiseen mallirakenteeseen, jonka tulos voidaan simuloida melko helposti esimerkiksi trendikäyristä saaduilla arvoilla tai kerätyllä ja aikaleimatulla mittaustiedolla. Kokeellisissa malleissa pohjalla on useimmiten differenssiyhtälö.

##### 4.1 Differentiaaliyhtälömallit

Differentiaaliyhtälö on yhtälö, joka kuvaa, miten yhden asian muuttuminen vaikuttaa toisen asian muuttumiseen (Glanville 2013, 112). Differentiaaliyhtälö on tuntemattoman funktion sisältävä yhtälö, jossa on muuttujan derivaattoja. Lisäksi haluamme selvittää tuntemattoman funktion, joko lausekkeena tai likimääräisesti. Differentiaaliyhtälön ratkaisuna tulee funktio, joka voidaan ajaa matemaattiseen ohjelmistoon tai graafiseen laskimeen. (Henttonen, Peltomäki & Uusitalo 2004, 94.)

$$y' + py = f(t) \tag{1}$$

$y'$                        $y$ -arvon derivaatta

$p$                          vakio

$f(t)$                     funktio ajansuhteen. (Henttonen, Peltomäki & Uusitalo 2004, 103)

$$y'' + ay' + by = f(t) \quad (2)$$

$y''$                        $y$ -arvon toinen derivaatta

$y'$                          $y$ -arvon derivaatta

$y$                           $y$ -arvo

$a, b$                     vakio. (Henttonen, Peltomäki & Uusitalo 2004, 109)

Differentiaaliyhtälöstä matemaattisena esimerkkinä toimivat kaavat 1 ja 2 Kyseessä on ensimmäisen ja toisen kertaluvun differentiaaliyhtälöt, sillä korkein derivaatta ( $y'$  ja  $y''$ ) on ensimmäistä ja toista astetta. Derivaattojen määrän näkee pilkkujen tai pisteiden määrästä kirjaimen yläpuolella.

Differentiaaliyhtälöt voidaan jakaa kahteen osaan:

- homogeenisiin
- epähomogeenisiin. (Henttonen, Peltomäki & Uusitalo 2004, 94.)

Homogeenisessa differentiaaliyhtälössä kaikissa termeissä on tuntematon funktio tai kyseisen funktion derivaattoja. Kaavassa 4 on ensimmäisen kertaluvun homogeeninen differentiaaliyhtälö. Heterogeenisessä differentiaaliyhtälössä on ainakin epähomogeeninen termi, joka ei sisällä etsittävää funktiota tai funktion derivaattaa. (Henttonen, Peltomäki, Uusitalo 2004, 95 - 96.) Kaavassa 3 on ensimmäisen kertaluvun epähomogeeninen differentiaaliyhtälö.

$$\frac{df(t)}{dt} = af(t) \quad (3)$$

a vakio. (Differential Equations s.a)

$$a \frac{df(t)}{dt} + bf(t) + c = 0 \quad (4)$$

a,b,c vakio. (First Order Non-homogeneous Differential Equation s.a)

Differentiaaliyhtälöt määritellään niiden kertalukujen mukaan. Kertaluku tarkoittaa korkeinta differentiaaliyhtälössä esiintyvää derivaattaa. Helpommissa differentiaaliyhtälöissä voi olla vain yksi derivaatta, kun taas vaikeammissa kaksi derivaattaa (derivaatan derivaatta). Mitä korkeampi kerta luku on, sen tarkemmin voidaan tehdä esimerkiksi viritys. Fysikaalisessa muodossa kertaluku on ikään kuin varasto, mitä enemmän kertalukuja sitä enemmän varastoja, kuten säiliöitä (Harju & Marttinen 2000, 72).

Huomioitavaa on differentiaaliyhtälöiden esitystavasta. Differentiaaliyhtälössä derivaatta esitetään termillä  $y'$ , toinen tapa esittää derivaatta on kaava 5. Kyseisessä yhtälössä  $dY$  ja  $dt$  edustavat differentiaalimuuttujia. Toki muuttujat voidaan ilmoittaa millä kirjainyhdistelmillä tahansa tai vaikkapa fysiikan yksikköinä, kuten esimerkiksi lämpötila  $T$ , aika  $t$ , matka  $s$ .

$$y' = \frac{dy}{dt} \quad (5)$$

Differentiaaliyhtälömalleja ei välttämättä tarvitse kokonaan luoda alusta loppuun, vaan esimerkiksi automaationalan ja mallintamisen kirjoista löytyy valmiita differentiaaliyhtälöitä tiettyjä prosesseja varten. Näiden valmiiden differentiaaliyhtälöiden kanssa pääsee hyvin alkuun. Kappaleessa 4.2 esiintyvää siirtofunktiomuotoa voidaan käyttää rakentaessa dynaamista mallia, mutta helppo keino tehdä Simulink-ohjelma differentiaaliyhtälöstä on kirjoittaa Simulink-ohjelman komponenttien avulla yhtälö.

## 4.2 Laplace-tason mallit

Laplace-muunnos (merkitään  $\mathcal{L}$ -muunnos, käänteinen Laplace-muunnos merkitään  $\mathcal{L}^{-1}$ ) on integraalimuunnos. Sen avulla lineaariset vakiokertoimiset differentiaaliyhtälöt pystytään muuttamaan algebrallisiksi yhtälöiksi. (Henttonen, Peltomäki & Uusitalo 2004, 127.). Laplace-muunnos ei vaikuta prosessin dynamiikkaan tai sen fysikaalisiin ominaisuuksiin, vaan ne pysyvät ennallaan sekä tehdessä Laplace- tai käänteistä Laplace-muunnosta. Laplace-muunnoksen tarkoituksena on ikään kuin pakata differentiaaliyhtälömme pienempään pakettiin, jotta tilaa säästyisi. Esimerkiksi jos tutkitaan säätimen kriteerejä, joudutaan Laplace-siirtofunktio muuttamaan differentiaaliyhtälöksi, jotta voimme suorittaa integraalilaskutoimituksen.

$$F(s) = \int_0^{\infty} f(t) e^{-st} dt \quad (6)$$

(Boyd s.a)

$$f(t) = \frac{1}{2\pi j} \int_{\sigma+j\infty}^{\sigma-j\infty} F(s) e^{+st} ds \quad (7)$$

(Boyd s.a)

Kaavassa 6 ja 7 on esitetty Laplace-muunnoksen integraalikaavat. Kaavassa 6 on Laplace-muunnoksen kaava ja kaavassa 7 on käänteisen Laplace-muunnoksen kaava. Kyseisten kaavojen mukaisia laskutoimituksia ei kuitenkaan tarvita, sillä matemaatikot ovat johtaneet valmiiksi tyypillisimmät Laplace-parit. Luettelo tyypillisimmistä Laplace-muunnoksista löytyy esimerkiksi Tekniikan Kaavastosta (Tekniikan Kaavasto 2005).

Differentiaaliyhtälön ja Laplace-muunnetun differentiaaliyhtälön ero on niiden aikatasoissa. Differentiaaliyhtälö on aikatasossa ja Laplace-muunnettu yhtälö on s-taajuusavaruudessa (Henttonen, Peltomäki & Uusitalo 2004, 128). Laplace-kaavoissa olevasta s-termi on oma terminsä ja yleensä s-termille ei tehdä mitään. Tietokone käsittelee s-termin sille ohjelmoidulla tavalla.

Laplace-siirtofunktio on Laplace-muunnoksen sovellus, jolla kuvataan systeemin tulon ja lähdön välistä riippuvuutta. (Henttonen, Peltomäki & Uusitalo 2004, 145). Siir-

tofunktio on tarkka kuvaus prosessista, mutta tiettyjen tarkkuuden rajoissa. Tarkka kuvaus prosessista kasvattaa siirtofunktion kokoa, mitä isompi siirtofunktio, sen vaikeampi sitä on tulkita. Vaikeat siirtofunktiot kannattaa sieventää ja mahdollisesti katsoa likimääräinen tulos. Kaikkein tehokkaimmin ja helpoiten siirtofunktiota pystyy käsittelemään tietokoneella.

Laplace-siirtofunktio kaavana esitettynä on:

$$G(s) = \frac{Y(s)}{U(s)} \quad (8)$$

$Y(s)$             lähtö Laplace-muodossa

$U(s)$             tulo Laplace-muodossa.

(Henttonen, Peltomäki & Uusitalo 2004, 145)

Kuten differentiaaliyhtälöitä myös Laplace-siirtofunktiomalleja on valmiina. Prosessitekniikan ja prosessien suunnitteluun liittyvissä kirjoissa, mallintamisesta kertovissa kirjoissa sekä internetlähteistä löytyy eri prosesseille valmiita siirtofunktioita. Valmiita siirtofunktioita löytyy mm. virtapiireille, DC-moottoreille, virtauksille jne. Kaikissa kirjoissa ei välttämättä aina ole merkitty differentiaaliyhtälöä, josta kyseinen siirtofunktio on tehty. Tämä vaikuttaa melkoisesti siihen, kuinka tarkka saamastamme mallista tulee, koska emme voi arvioida, mitä asioita yhtälöstä on karsittu pois. Tärkeimmät siirtofunktiomuodot ovat ensimmäisen kertaluvun ja viiveen siirtofunktio ja integroivan prosessin siirtofunktio.

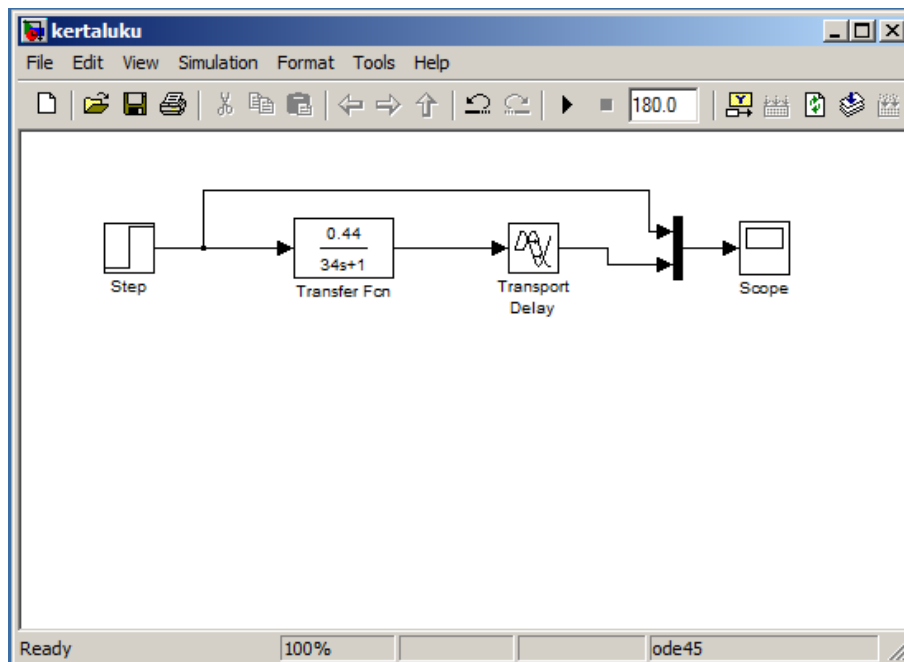
Ensimmäisen kertaluvun ja viiveen siirtofunktio Laplace-muodossa on esitetty kaavassa 9. Ensimmäisen kertaluvun ja viiveen siirtofunktiomallista voidaan laatia Simulink-ohjelma (kuva 13), yhtälö osuuden muodostaa siirtofunktio lohko ja viive, jonka avulla saadaan kuvassa 14 oleva askelvaste. Kuvassa 14 keltainen viiva on ohjaukseen tehty askel ja violetti on prosessissa tapahtunut muutos.

$$G(s) = \frac{Y(s)}{U(s)} = \frac{K \cdot e^{-sL}}{Ts+1} \quad (9)$$

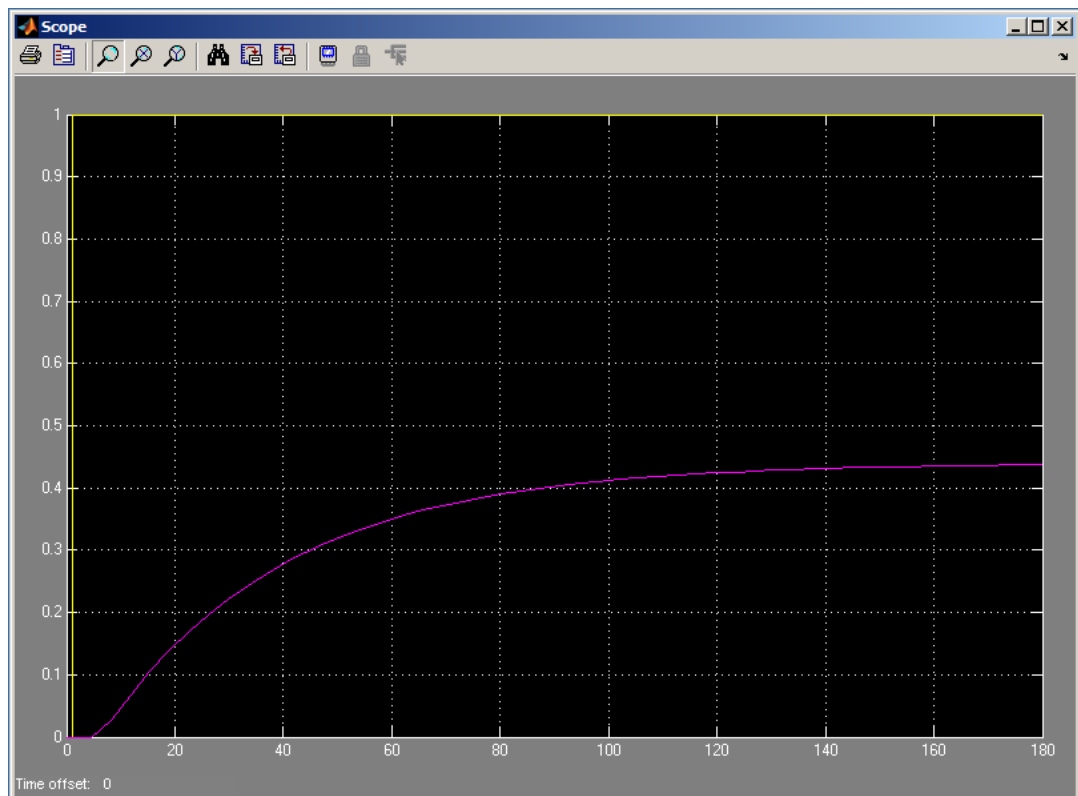
$K$             prosessin vahvistus

T aikavakio

L viive. (Harju & Marttinen 2000, 117)



Kuva 13. Ensimmäisen kertaluvun ja viiveen siirtofunktiota vastaava Simulink-ohjelma

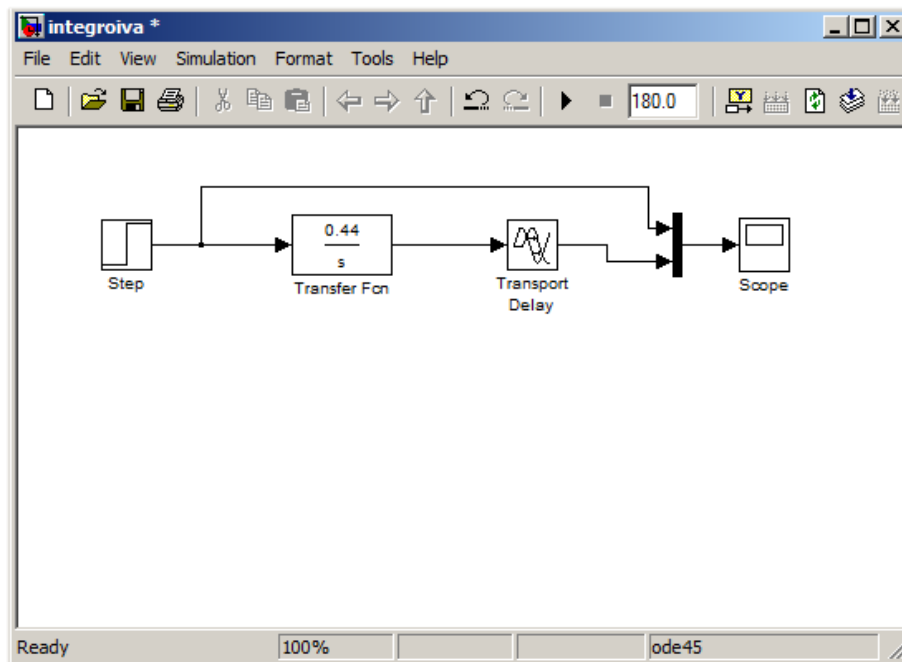


Kuva 14. Ensimmäisen kertaluvun ja viiveen askelvaste

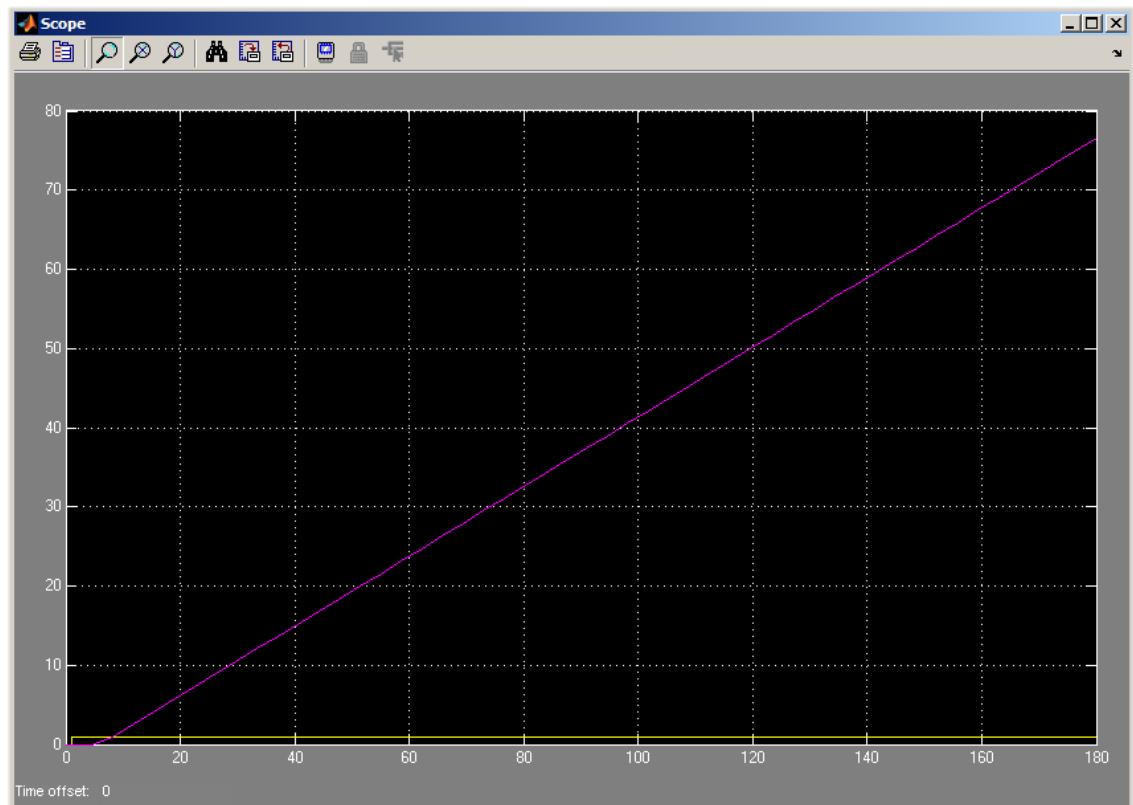
Toinen siirtofunktio on integroivan dynaamisen systeemin siirtofunktio. Se on esitetty kaavassa 10. Kaavan avulla saadaan rakennettua Simulink-ohjelma (kuva 15), jossa prosessin dynamiikka on siirtofunktio lohkoissa ja viivelohkoissa, josta saadaan kuvan 16 mukainen askelvaste.

$$G(s) = \frac{K \cdot e^{-Ls}}{s} \quad (10)$$

(Harju & Marttinen 2000, 81)



Kuva 15. Integroivan prosessin siirtofunktio Simulink-ohjelma



Kuva 16. Integroivan prosessin askelvaste

Siirtofunktion kertaluvulla tarkoitetaan prosessin hitautta. Kertaluvulla tarkoitetaan siirtofunktioiden kohdalla, kuinka monta hitautta aiheuttavaa tekijää prosessissa on. Prosessitekniikassa hitaudet pääasiassa tulevat säiliöistä. Liitteessä 1 ja 2 on esitetty tyypillisimpiä 1. ja 2. kertaluvun prosesseja. Yleensä prosessit kuitenkin käyttäytyvät 1. kertaluvun mukaisesti.

### 4.3 Differenssiyhtälömallit

Differenssiyhtälöitä käytetään, kun kyseessä on näytteenottotieto eli tietyin aikavälein suoritettu mitta. Differenssiyhtälön muuttujan tila poikkeaa dynaamisesta mallin muuttujan tilasta siten, että tila on sama niin kauan, kunnes uusi näytteenotto sanoo, että tila on muuttunut. Esimerkiksi jos näytteenottojen välillä olisi aika 5 sekuntia ja näytteessä tapahtuisi muutos 3 sekunnin kuluttua edellisestä näytteen otosta, olisi yhä edelleen käytössä aikaisempi tila. Tämä näkyy esimerkiksi kuvissa 17 ja 18, jossa jakava-aikainen massavirran virtaus on muutettu näytteenottoon perustuvaksi.

Kaavassa 11 on esimerkki yksinkertaisesta differenssiyhtälöstä, jossa uusin ulostulo  $y(k)$  määräytyy edellisestä ulostulosta  $y(k-1)$ .



$$y(k) = y(k - 1) + u(k) \quad (11)$$

(Lyytikäinen & Mäkelä 2013)

Aikasarjamalleiksi kutsutaan malleja, joissa on riippuvuus kahden suureen välillä. Diskreeteissä eli digitaalisissa aikasarjamalleissa parametrit riippuvat näytteenottovälistä. (Harju & Marttinen 2000, 82 - 83.)

Aikasarjamallien mallintamiseen on useita eri vaihtoehtoja:

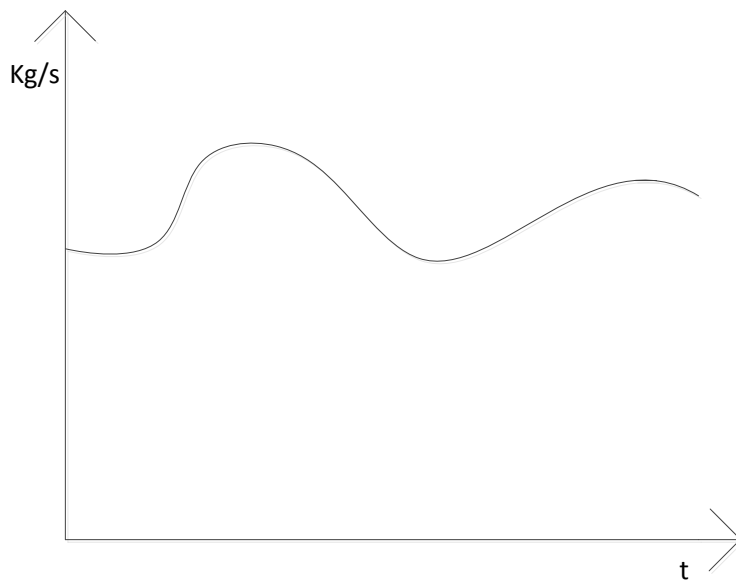
- ARX - AutoRegressive with eXogenous input
- ARMAX - Auto-Regressive Moving Average with Exogenous Input
- ARIMAX - Auto-Regressive Integrated Moving Average with Exogenous Input
- NARMAX -Non-linear Time Series Models (Tham 2000)

Aikasarjamallien luomiseen tärkeimmät menetelmät ovat ARX ja ARMAX. ARX-mallissa luodaan malli kahden muuttujan tiedoista esimerkiksi virtausnopeuden ja pumpun pyörimisnopeuden tiedoista. ARX-mallissa tiedonkeruu tietojen täytyy olla samalta ajalta. Matlab-ohjelmassa ARX-mallin kelposuutta arvioidaan vertaamalla mallia, mittaustulosten kanssa. ARX-mallista saadut kertoimet voidaan sijoittaa Simulink-ohjelmassa olevaan Z-siirtofunktiolohkoon ja näin voimme testata, miten prosessi käyttäytyy eri arvoilla. ARX-mallista on olemassa myös ARMAX-versio, joka ottaa huomioon Mittauskohinan, kuitenkin useimmiten riittää pelkkä ARX-algoritmi, joka esiintyy kaavassa 12.

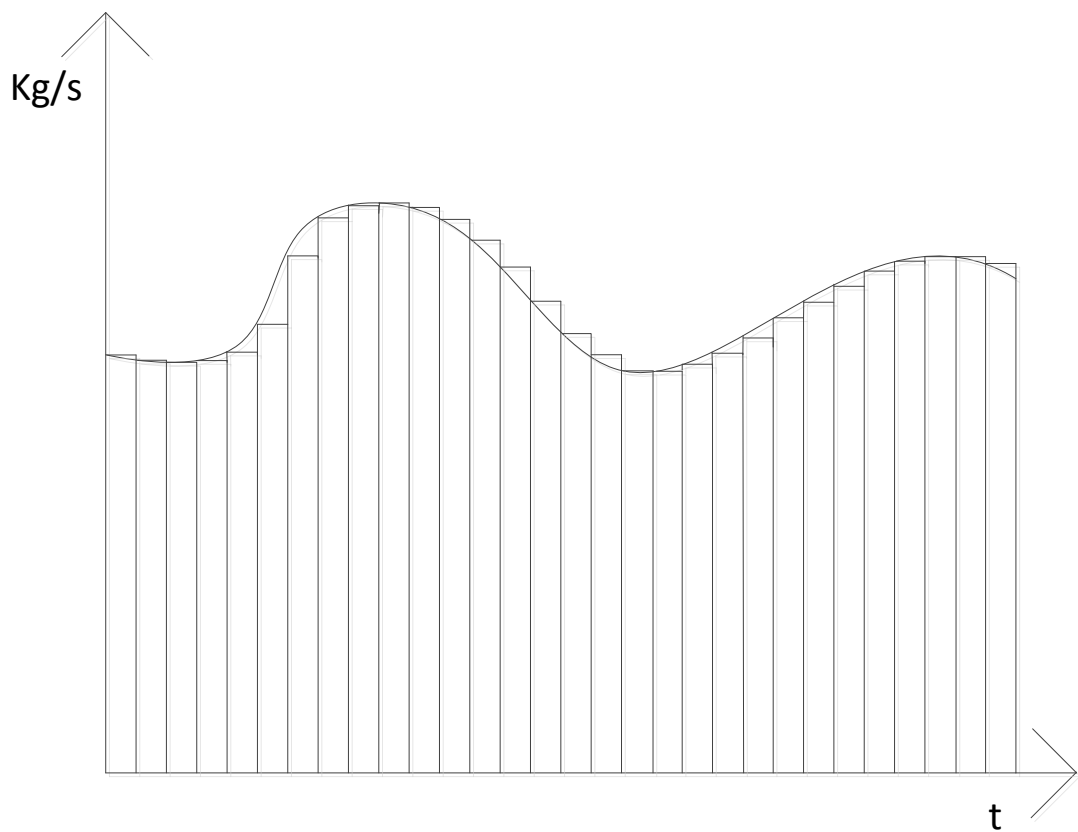
Algoritmimuodossa ARX-malli on:

$$y(k) = a_1y(k - 1) + \dots + a_{na}y(k - na) = b_1u(k - 1) - \dots - b_{nb}u(k - nb) + V_k \quad (12)$$

(Harju & Marttinen 2000, 84 )



Kuva 17. Massavirran muuttuminen esitettynä jatkuva-aikaisena mallina



Kuva 18. Massavirran muutos esitettynä diskreettinä mallina

#### 4.4 Z-tason mallit

Z-tason malleissa idea on sama, kuin Laplace-muunnoksissa: pyrimme saamaa differenssiyhtälön pienempään muotoon, jotta sitä olisi helpompi muokata. Differenssiyhtälö ei ole jatkuva, joten Laplace-muunnosta ei voida soveltaa tähän. Z-siirtofunktio on Z-muunnoksen sovellus. Z-muutosta käytetään diskreettiaikaisissa järjestelmissä, jolloin signaalit ovat pulssijonoja. Nämä pulssijonot voidaan ajatella esimerkiksi palkkeina kuten kuvassa 18.

$$X(z) = \sum_{n=0}^{\infty} x(n)z^{-n} \quad (13)$$

(Råde & Westergen 2004, 327 )

$$x(n) = \frac{1}{2\pi i} \int X(z)z^{n-1}dz = \frac{1}{n!} \left( \frac{d}{dz^{-1}} \right)^n X(z) \Big|_{z^{-1}=0} \quad (14)$$

(Råde & Westergen 2004, 327)

Kaavoissa 13 ja 14 on määritelty Z-muunnos. Kaavassa 13 on Z-muunnos ja kaavassa 14 on käänteinen Z-muunnos. Z-siirtofunktiomalleja käsitellään samaan tapaan kuin Laplace-siirtofunktiomalleja. Z-muunnoksille löytyy omat muunnostaulukkonsa esimerkiksi kirjassa *Modern Control Systems* (Dorf & Bishop 2011, 1014 - 1015).

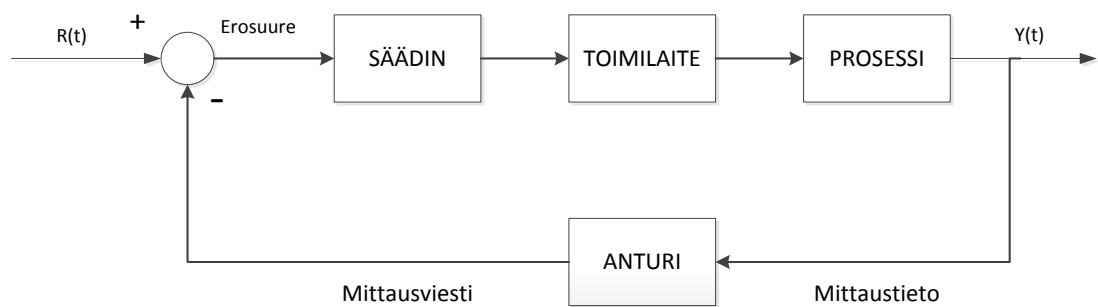
#### 4.5 Lohkokaavioesitys

Prosessin kulkua voidaan myös ajatella lohkokaavioiden avulla. Lohkokaavioiden avulla seurataan signaalien kulkua prosessissa ja, miten signaalit vaikuttavat muihin osa-alueisiin. Signaalit voivat esimerkiksi kuvastaa prosessin mittausta tai prosessin säädintä. Matlabin Simulink-ohjelmassa systeemit käsitellään lohkokaavioiden kautta. Helpoin tapa tutustua lohkokaavioihin on ajatella lohkokaaviot säätöpiirinä, jossa signaali on standardi virtaviesti 4...20 mA.

Kuvassa 19 on suljetun säätöpiirin toiminta. R(t) on asetusarvo, joka menee säätimelle. Säädin ajaa sen hetkisten tietojen perusteella asetusarvon PID-algoritmin läpi ja antaa ohjausarvon toimilaitteelle, joka voi olla tässä tapauksessa esimerkiksi venttiili. Venttiilin ohjaus vaikuttaa prosessiin, josta tulee tieto anturille. Anturilta tieto menee

summaimeen, jossa erotus tapahtuu. Erotuksesta jäävä erosuure menee säätimelle, jossa säädin muuttaa ohjausta tilanteen mukaan.

Systeemin koostuessa useammasta pienemmästä osaprosessista, muuttuu myös lohko-kaaviokuvaus vaikeaksi. Lohkokaavioita sieventämällä on tarkoituksena päästä lohko-kaaviossa siihen tilaan, että käytössä on vain yksi siirtofunktiolohko, joka sisältää kaikkien osaprosessien siirtofunktiot. (Savolainen & Vaitinen 1995, 155.) Lohkokaavioiden sieventämisessä käytetään siihen tarkoitettu sääntöjä. Lohkokaavion sievennys säännöt ovat mutta kokonaisuuden ollessa pienempi, ei välttämättä tarvita sievennystä. Lisäksi prosessin lohko-kaaviossa ei tarvitse sieventää kaikkea, vaan ne termit, jotka aiheuttavat sotkuisuutta lohko-kaaviossa.



Kuva 19. Suljetun säätöpiirin toiminta esitettynä lohkokaaviomallina (Dorf & Bishop 2011, 25)

#### 4.6 Viiveiden käsittely

Mallintamisessa meille voi tulla tilanteita, jossa emme pysty arvioimaan tai käsittelemään viivettä. Viiveiden käsittelyyn on olemassa erilaisia lähestymistapoja, jotka jakautuvat simulointiin ja käytäntöön. Simuloinnissa voi olla tapauksia, jossa viiveiden arviointi on vaikeaa. Tällaisissa tapauksissa olisi suositeltavaa kokeilla Padé-approksimaatiota. Käytännön viiveellisissä prosesseissa olisi kannattavinta käyttää prediktiivisiä säätimiä. Prediktiivistä säätimistä tärkeimmät ovat Smith-prediktori ja Kalman-suodatin.

$$L(s) = e^{-\tau s} \quad (15)$$

(Dorf & Bishop 2011, 693)

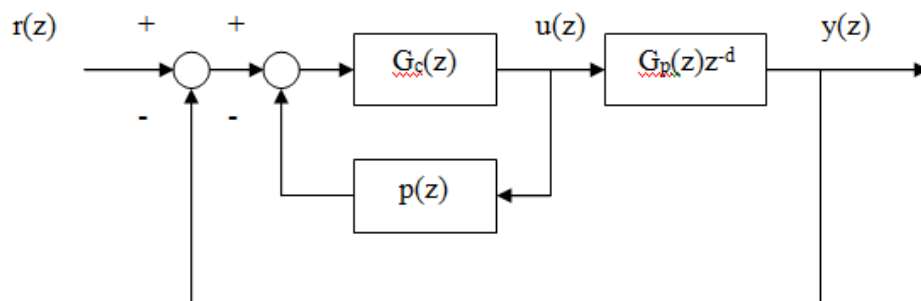
Padé-approksimaation ideana on luoda siirtofunktio viiveelle, jonka siirtofunktio esiintyy kaavassa. Padé-approksimaatiota käytetään silloin, jos ei normaalisti ole mahdollista käyttää siirtofunktiota. Padé-approksimaatiossa esiintyy pieni virhe, mutta tämä ei loppupeleissä vaikuta paljoakaan viiveen arvioitiin. Padé-approksimaatio on yksikertainen, mutta toimiva matemaattinen keino. Kuten kaavasta 16 huomataan osoittajassa ja nimittäjässä ei ole varsinaisesti muuta eroa, kuin etumerkki.

Padé-approksimaation funktio on:

$$e^{-\tau s} \approx \frac{1 - k_1 s + \dots + k_n s^n}{1 + k_1 s + \dots + k_n s^n} \quad (16)$$

(Haugen 2009)

Smith-prediktorin vaatii toimiakseen viiveellisen prosessin takaisinkytkennän siirtofunktiot. Säättäjän ulostulon vasteen ja takaisin kytkennän kautta voidaan laskea eli ennakoita ja vähentää tulos erosignaalista. Lisättäessä tähän sama signaali, josta viive on poistettu, viiveen vaikutus saadaan lähestulkoon eliminoitu. Smith-prediktori parantaa systeemin stabiiliusominaisuuksia, mutta se ei kuitenkaan poista kokonaan systeemin viivettä. (Hietanen, 1997.) Kuvassa 20 on Smith-prediktori esitettynä lohkokkaaviomuodossa ( $G_c$  on PID-säädin,  $p$  on siirtofunktio ja  $G_p$  on siirtofunktio viiveen kanssa). Lohkokaavioesityksestä tulee huomata, että kyseessä on diskreettiversio Smith-prediktorista.



Kuva 20. Smith-prediktorin lohkokkaavioesitys diskreettinä (Hietanen, 1997)

#### 4.7 Prosessikokeet ja sopivan mallin valitseminen

Olemassa oleva prosessin mallintamisessa hyödynnetään prosessikokeita. Prosessikokeella selvitetään käytännössä, miten muutos vaikuttaa prosessin käytökseen. Prosessin käyttäytymisen selvittäminen on tärkeää, jotta pystymme hallitsemaan prosessia. Prosessikokeista yleisimmät ovat askelkoe ja satunnaispulssikoe. Prosessikokeiden suorittaminen vaatii suunnittelua, aikaa ja ne täytyy suunnitella siten, että prosesseihin tehdyt muutokset eivät aiheuta vaaraa ihmisille tai ympäristölle.

Askelkokeessa säädin laitetaan käsiajolle, jonka jälkeen poistetaan mahdollisesti piirissä olevat kuormitushäiriöt eli odotetaan, että prosessi on tasainen. Tämän jälkeen tehdään säätimen lähtöön askelmainen muutos, jossa ohjataan esimerkiksi venttiili 20 % auki. Askelmaisen muutoksen tekemisen jälkeen odotetaan, että prosessi tasaantuu ja tallennetaan sekä mittauksen, että ohjauksen trendikäyrät esimerkiksi tietokoneen kiintolevyille. Annetun askeleen pitää olla teräväkulmainen, jotta askelkoe onnistuu. Askelkokeen trendikäyrän perusteella voimme päätellä, onko kyseessä integroivaprosessi vai ensimmäisen kertaluvun ja viiveen funktio. Askelkokeella pääsemme Laplace-siirtofunktiomalliin. (Harju & Marttinen 2000, 96)

Satunnaispulssikokeessa tehdään askelia niin ylös- että alaspäin ja samaan aikaan kerätään tietoa tekemistämme muutoksista dataloggeriin. Ennen dataloggerin hankkimista kannattaa varmistaa, mitä tiedostomuotoa käytössä oleva matemaattinen ohjelmisto tukee. Esimerkiksi Matlab-ohjelmistoon voi syöttää Microsoftin Excel-ohjelman muodossa olevaa dataa. Askelten tulee olla samankokoisia sekä ylös että alaspäin ja samalla tiedonkeruun tiedon täytyy olla samalta ajalta. Pulssien pituus sekä kesto arvotaan satunnaisesti eli välillä tulee pidempiä tai lyhyempiä pulsseja. Näitä pulssijonoja kopioidaan ja syötetään tarpeeksi useasti peräkkäin. Satunnaispulssikoe voidaan suorittaa siten, että säädin on joko automaattilla tai käsiajolla, mutta parhaan mahdollisen tuloksen saamiseksi suositellaan käsiajoa. Satunnaispulssikokeesta saadulla tiedolla voidaan tehdä esimerkiksi ARX-malli prosessista. Satunnaispulssikoe vie huomattavasti enemmän aikaa, kuin askelkoe. (Harju & Marttinen 2000, 97)

Prosessikokeista askelkoe on mahdollista tehdä myös teoreettisesti. Tämä onnistuu kertomalla siirtofunktio 1:en Laplace-muunnoksella ja tekemällä käänteisen Laplace-muunnoksen. Käänteismuunnoksen jälkeen saadusta differentiaaliyhtälöstä voidaan analysoida esimerkiksi tasaantuuko prosessi, kuinka nopeasti prosessi tasaantuu, esiin-

tyykö värähtelyä jne. Kuitenkin helpoin tapa testata teoreettista mallia on käyttää esimerkiksi Matlabin Simulink-ohjelmaa ja simuloida siirtofunktion toiminta tai tehdä käänteismuunnos Mathcadin avulla.

## 5 SÄÄTIMIEN MALLINTAMINEN

Säätöpiiriin kuuluu säädettävä prosessi toimilaitteiden kanssa, mittausanturi- ja lähetin ja säädin. Säätöpiirin avulla hallitaan prosessia, ylläpidetään prosessin tasapainotilaa ja hallitaan prosessia tasapainotilan muutoksen aikana. (Harju & Marttinen 2000, 13).

Säätimen tärkeimmät tehtävät ovat:

- turvallisuus
- ympäristön suojele
- välineistön suojele
- pehmeä operointi
- laadun säilyttäminen
- seuranta & diagnosointi. (Marlin 1995, 22 - 26.)

Perinteisesti ohjausviesti on 4...20 mA-virtaviesti. (Harju & Marttinen, 13) Uudemmissa teollisuuden laitoksilla on siirrytty kenttäväyläteknikkaan, jossa ohjausviestit kulkevat digitaalisena binäärikoodina. (Mäkelä 2013) Jotta säädettävä suure saadaan toimimaan halutulla tavalla erilaisissa asetusarvon ja kuormitushäiriöiden muutostilanteissa, on prosessin käyttäytyminen tunnettava. (Harju & Marttinen 2000, 14)

PID-säätimen lisäksi on olemassa muita mallinnus- ja säätötapoja. Sumea logiikka on säätötapa, joka pyrkii pääsemään eroon automaation raja-tyyppisestä ajattelusta. Perinteisesti automaatiossa mennään ylä- ja rajatyypisellä säädöllä mutta sumea logiikka pyrkii ottamaan huomioon myös välimuodot. (Mäkelä & Lyytikäinen 2013)

Neuroverkkojen ideana on luoda tekoäly, joka oppii. Se koostuu pääosin yksinkertaisista solmuista, joiden tarkoituksen on olla keinotekoisia hermosoluja. Hermosolujen välillä on liitoksia, jotka vastaavat aivojen synapseja. Neuroverkossa tietoa käsitellään liitoksien avulla koodilohkon leviämisenä eri solmuissa. Neuroverkko voidaan opettaa annetuilla esimerkeillä, mutta sitä ei kuitenkaan ohjelmoida suoraan toi-

mimaan ennalta määritetyillä tavoilla vaan sen tarkoituksena on oppia eri tapauksista. (Honkela 1998)

### 5.1 PID-säätimien toiminta ja algoritmit

PID-säädin koostuu kolmesta osasta. P-osa suorittaa kertomisen, I-osa suorittaa integroinnin ja D-osa suorittaa derivoinnin (Ristola 2013). Kuvassa 19 voidaan nähdä PID-säätimen toiminta: käyttäjä antaa asetusravon R, summaimessa tapahtuu erotus prosessin mittausarvon kanssa ja erosuure menee säätimelle, jolloin PID-algoritmi laskee toimilaitteelle ohjauksen. PID-säätimestä voidaan käyttää myös variaatioita P-,PI-,PD-säätimiä, näistä suosituin prosessiteollisuudessa on PI-säädin. (Harju & Marttinen, 50). PID-säätimen variaation voi selvittää helposti avaamalla prosessinohjausjärjestelmästä piiri-ikkunan ja katsomalla parametrit kohtaa, jos jokin parametri on 0, silloin se ei ole käytössä. PID-algoritmi laskee yhteen kaikki kolme osa-alueita. Kaavamuotoja kyseiselle algoritmille on kaksi kappaletta, joista ensimmäinen on:

$$u(t) = K_p \left[ e(t) + \frac{1}{T_i} \int_0^t e(t) dt + T_d \frac{de(t)}{dt} \right] \quad (17)$$

$K_p$                       säätimen vahvistus

$e(t)$                      erosuure

$T_i$                         integrointiaika

$T_d$                         derivointiaika. (Harju & Marttinen 2000, 45 - 48)

Teollisuudessa PID-algoritmi ei ole samanlainen, kuin kaava 17. Teollisuuden käyttökohteita varten PID-algoritmi joudutaan diskretisoimaan, koska ideaalikaavassa (kaava 17) integraalia ja derivaattaa ei voida laskea suoraan. Tämä johtuu siitä, että ohjaussignaalin arvo muuttuu kokoajan, jolloin tarvitaan numeerinen likimääräiskäsittely. PID-algoritmi voidaan diskretisoida suorakulma-approksimaation avulla. Suorakulma-approksimaatiossa jatkuva-aikainen funktio jaetaan suorakulman muotoisiksi alueiksi joiden leveys on näytteenottoväli.

Ensimmäinen diskreetti versio, jossa lisätään mukaan näytteenottoaika, kuten kuvassa 18.



$$u_n = K_p(e_n + \frac{1}{T_i} \sum_{p=1}^n e_p \cdot T + T_d \cdot \frac{e_n - e_{n-1}}{T}) \quad (18)$$

$u_n$	ohjaussignaalin arvo
$e_n$	tämän hetkinen näytteenottoarvo
$e_{n-1}$	edellinen näytteenottoarvo
$e_p$	palkin korkeus
$T$	näytteenottoaika
$T_i$	integrointiaika
$T_d$	derivointiaika
$K_p$	säätimen vahvistus. (Lyytikäinen & Mäkelä 2013)

Pudotamme näytteenottonumeroa n yhdellä:

$$u_{n-1} = K_p(e_{n-1} + \frac{1}{T_i} \sum_{p=1}^{n-1} e_p \cdot T + T_d \cdot \frac{e_{n-1} - e_{n-2}}{T}) \quad (19)$$

$e_{n-2}$	aikaisempi näytteenottoarvo. (Lyytikäinen & Mäkelä 2013)
-----------	----------------------------------------------------------

Vähennämme puolittain:

$$u_n - u_{n-1} = K_p(e_n - e_{n-1} + \frac{1}{T_i} e_n \cdot T + T_d \frac{e_n - 2e_{n-1} + e_{n-2}}{T}) \quad (20)$$

$u_{n-1}$	aikaisempi ohjausarvo (Lyytikäinen & Mäkelä 2013)
-----------	---------------------------------------------------

Diskretisoidun PID-algoritmin viimeiseksi muodoksi saadaan:

$$u_n = u_{n-1} + K_p \left( e_n - e_{n-1} + \frac{T}{T_i} e_n + \frac{T_d}{T} (e_n - 2 \cdot e_{n-1} + e_{n-2}) \right) \quad (21)$$

(Lyytikäinen & Mäkelä 2013)

PID-säätimen K-kerroin eli vahvistus vaikuttaa ohjauksen ja erosuureen välillä suhdetoimena. Vahvistuksen arvoa nostamalla voidaan nopeuttaa järjestelmän käyttäytymistä, mutta useimmiten  $K_p$ -arvoa ei laiteta kovinkaan suureksi sillä liian korkea  $K_p$ -arvo saa säätimen värähtelemään. (Harju & Marttinen 2000, 50.)

I-termillä tarkoitetaan integrointiaikaa. Integrointiaika hyödyntää prosessin historiatietoja sekä aikaisempia erosuureen arvoja. Integrointiajan päätarkoituksena on, että se poistaisi jatkuvuus tilan virheet sillä pienikin erosuureen arvo kasvattaa lähtöä integrointiosan avulla. Mitä suurempi I-termi on, sitä pienempi vaikutus I-termillä on säätimessä myös epästabiilimpi ja värähtely alttiimpi säädin on, samalla poistuu säätöpoikkeamaa. Pieni integrointiaika aiheuttaa suuren kasvunopeuden. (Harju & Marttinen 2000, 50.)

D-osa muodostaa ohjauksen erosuureen tai tarkasteltavan säädettävän suureen muutosnopeuden pohjalta. Muutosnopeus tavallaan ennustaa järjestelmän käyttäytymistä. Mitä suurempi derivointiaika on, sitä suurempi vaikutus D-osalla on säätimessä. Prosessin sisältäessä hitautta, pitää derivointiaikaan suhtautua varovaisesti. D-osa korostaa korkeita taajuuksia, kuten mittauskohinaa, tämän takia täytyy käyttää joko signaalin suodatusta tai laskea derivaatta eri tavalla. Tämän takia prosessiteollisuudessa onkin harvemmin käytössä D-osa. Mekaanisissa järjestelmissä D-osaa käytetään enemmän, koska siellä signaalit siirtyvät lyhyempiä reittejä. (Harju, Marttinen 2000, 50.)

### 5.3 Säätimen viritys

Säätimen täytyy olla hyvin viritetty, että siitä saataisiin hyötyä (Vaattinen & Savolainen, 103). Esimerkiksi säädin on huonosti viritetty, jos operaattori joutuu ohjamaan säätöpiiriä useasi käsiohjauksella (Mäkelä 2013). Säätimen virityksessä tärkeimmät parametrit, jotka tulee ottaa huomioon, ovat aikavakio ja viive. Otetaan esimerkiksi säädin jossa on PID-algoritmi. PID-algoritmi vaatii toimiakseen jonkinlaiset numeroarvot, jotta se toimii, mutta numeroarvot eivät voi kuitenkaan olla sattumanvaraisesti valittuja, koska säädin ei tiedä mitään prosessista. Väärin tehty viritys voi aiheuttaa sen, että säädin jää vain värähtelemään. Tämän takia viritys täytyy tehdä oikein.

Aikavakiolla tarkoitetaan nopeutta, jolla energia- tai ainevarastot muuttuvat, kun tapahtuu muutoksia. Mitä suurempi aikavakio on, sitä hitaampi järjestelmä on (Savolainen & Vaattinen 1995, 25) Aikavakio lasketaan muutostilan alkamisesta siihen asti, kunnes ollaan 63 % kohdalla lopullisesta muutoksen arvosta. (Savolainen & Vaattinen 1995, 27)

Viiveellä (kuollut aika tai horrosaika) tarkoitetaan ajanjaksoa jonka aikana systeemi ei reagoi herätteeseen tai järjestelmässä tapahtuneeseen muutokseen. (Savolainen & Vaattinen 1995, 27) Prosessissa olevaa viivettä ei voida poistaa välttämättä kokonaan, mutta viiveellisen prosessin käsittelyssä voidaan ottaa avuksi prediktiiiviset säädöt, joita käsiteltiin kappaleessa 4.6.

Lambda-iritys on erityisesti PI-säädölle ja 1. kertaluvun ja viiveen prosessille kehitetty viritysmenetelmä. (Lyytikäinen & Mäkelä 2013) Lambda-irityksen ideana on, että valitaan jo iritys vaiheessa säätöpiirin nopeuden parametri  $\lambda$  (lambda), jonka arvo voidaan valita väliltä 0,5 - 5 (Harju & Marttinen 2000, 118). Lambda-iritystä varten prosessiin tarvitsee tehdä askelkoe. Virityksen K, T ja L -arvot saadaan arvioitua askelkokeesta saadun trendikäyrän avulla. Olen kokeillut lambda-iritystä säätöpiirien simuloinneissa ja tulokset ovat aina olleet hyviä. Lambda-iritys on helppo viritysmenetelmä, jossa pärjätään aika pitkälti kahdella kaavalla.

Ensimmäisen kertaluvun viiveellisen prosessin iritys voidaan suorittaa kaavoilla 22 ja 23. Kaavalla 22 määritetään prosessin integrointiaika ja kaavalla 23 määritetään säätimen vahvistus. Nämä kaavat ovat tarkoitettu erityisesti PI-säätimille, jotka ovat prosessiteollisuudessa yleisimpiä säätimiä. Integroiville eli epästabiileille prosesseille

on myös oma virityskaavansa. Integroivan prosessin säätimille käytetään kaavoja 24 ja 25. Kaavassa 24 määritetään derivointiaika ja kaavassa 25 säätimen vahvistus. Kuten huomaamme integroivan prosessin säätämisessä käytetään PD-säädintä.

$$T_i = T \quad (22)$$

T prosessinaikavakio

T<sub>i</sub> integrointiaika. (Harju & Marttinen 2000, 119)

$$K_p = \frac{1}{K(\lambda + \frac{L}{T})} \quad (23)$$

K<sub>p</sub> säätimenvahvistus

λ säätimen aikavakio

K suhteellinen vahvistuskerroin

L viive. (Harju & Marttinen 2000, 119)

$$T_d = T \quad (24)$$

(Harju & Marttinen 2000, 121)

$$K_p = \frac{1}{K(\Lambda + L)} \quad (25)$$

Λ säätimen aikavakio. (Harju & Marttinen 2000, 121)

Kappaleessa 6 olevissa esimerkeissä joudutaan tekemään muutama lisälaskutoimitus, koska Matlabin Simulink-ohjelman PID-säädinlohko toimii hieman eritavalla, kuin standardi PID-säädin. Esimerkiksi kun olemme saaneet laskettua lambda-virityksellä parametrit, ne joudutaan laskemaan vielä kerran kaavoilla 24, 25 ja 26.

$$\text{Proportion} = \text{Säätimen vahvistus} = K_p \quad (26)$$

$$\text{Integral} = \text{Integrointiaika} = \frac{K_p}{T_i} \quad (27)$$

$$\text{Derivative} = \text{Derivointiaika} = \frac{K_p}{T_d} \quad (28)$$

## 6 DYNAAMISTEN PROSESSIEN MALLINTAMINEN JA SIMULOINTI

Prosessin matemaattisen mallin rakentamista on helpottamassa energia-alaan liittyvä kirjallisuus, josta voi helposti saada valmiita mallirakenteita. Lisäksi internet tarjoaa ison kirjon erilaisia valmiita malleja. Useimmiten internetlähteistä löytyvät valmiit pohjautuvat mallintamiseen ja säätötekniikkaan liittyviin kirjoihin.

Matemaattisten mallien rakentaminen ja simulointi ei ole aina helppoa, vaan välillä voi tulla ongelmia itse prosessin tai matemaattisten ohjelmien kanssa. Tässä kappaleessa käydään läpi teoreettinen ja kokeellinen mallintaminen ja tullaan huomaamaan mahdollisia ongelmien aiheuttajia, jotka vaikeuttavat tai jopa estävät mallin luomisen.

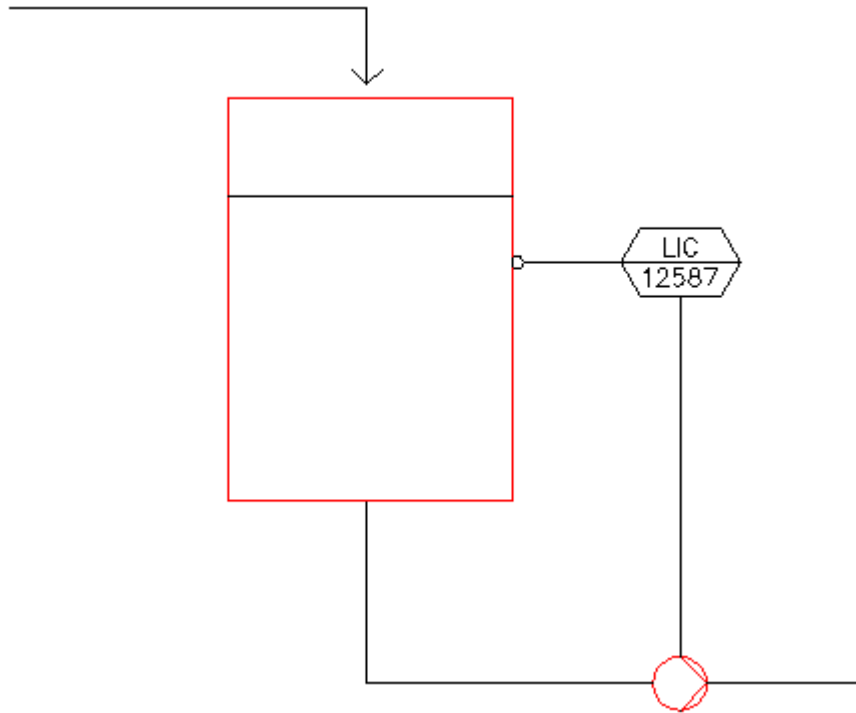
### 6.1 Teoreettinen mallintaminen luonnonlakeja hyödyntäen

Teoreettinen malli luodaan tapauksissa, kun esimerkiksi prosessia ollaan kehittelemässä tai luomassa. Malli luodaan hyödyntämällä fysiikan teorioita ja lakeja. Täytyy kuitenkin muistaa, että teoreettisen mallin pohjana on ideaaliprosessi eli malli ei välttämättä kaikilta osin vastaa oikeaa prosessia, mutta pääsemme kuitenkin riittävään tarkkuuteen.

Luonnonlakimallien pohjana ovat differentiaaliyhtälöt. Vaikka Laplace-muutoksella saadaankin yhtälö pienempään tilaan, on myös toinen helppo lähestymistapa. Kirjoittamalla differentiaaliyhtälö ”takaperin” voidaan matemaattinen malli saada aikaan helposti ilman Laplace-muunnosta. Tämän menetelmän huonopuoli on, että se ei ole yhtä selkeä, kuin Laplace-muunnos, mutta sillä pystytään näkemään, miten systeemi tulee käyttäytymään ja mikä suure vaikuttaa mihinkin. Käytän tätä metodia esimerkeissä

### 6.1.1 Säiliökapasiteetti integraattorimallina

Kuvassa 21 on ensimmäinen esimerkki prosessimme. Kyseessä on yksimuuttuja malli, jossa tuloveden ja poistoveden virtaus vaikuttavat pinnankorkeuteen. PI-kaaviosta voidaan jo hieman päätellä prosessin toimintaa. Mikäli vettä tuodaan enemmän kuin sitä poistetaan, veden pinta nousee, mikäli vettä pumpataan enemmän, kuin säiliöön tulee vettä, säiliö imeytyy tyhjäksi.



Kuva 21. Prosessin PI-kaavio

Seuraavaksi meidän tulisi etsiä kiinnostava suure, joka tässä tapauksessa on pinnan-  
korkeus. Prosessin staattinen tase eli tasapainotilan virtaustase on:

$$f_{tuleva} = f_{lähtevä} \quad (29)$$

$f_{tuleva}$  tuloveden virtausnopeus

$f_{lähtevä}$  lähtevän veden virtausnopeus. (Lyytikäinen & Mäkelä 2013)

Yhtälön perusteella prosessi on tasapainotilassa, kun tulovirtaus ja lähtövirtaus ovat yhtä suuret. Käytämme tässä prosessissa virtauksen suureena tilavuusvirtaa eli, kuinka monta kuutiota vettä virtaa sekunnissa. Toinen tapa ilmaista veden virtausmäärä olisi massavirta eli, kuinka monta kiloa sekunnissa vettä virtaa, mutta tässä tapauksessa massa virta ei ole järkevä suure. Tämä johtuu siitä, että meitä kiinnostaa veden määrä ja tilavuusvirrasta saadaan virtaus muutettua todella helposti litroksi. Massavirtaa käytetään enemmän esimerkiksi turbiinien mekaanisen tehon määrittämisessä annettujen prosessitietojen perusteella. Suureina ilmaistuna saadaan aikanaan yhtälö:

$$[f_{tuleva}] = [f_{lähtevä}] = \frac{m^3}{s} \quad (30)$$

(Lyytikäinen & Mäkelä 2013)

Dynaamisen taseen määritelmä prosessissamme on: Tilavuuden muutos aikayksikössä  
= tuleva tilavuusvirtaus – lähtevä tilavuusvirtaus

$$\dot{V} = f_{tuleva} - f_{lähtevä} \quad (31)$$

$\dot{V}$  tilavuudenmuutos aika yksikössä. (Lyytikäinen & Mäkelä 2013)

$$\frac{dV}{dt} = f_{tuleva} - f_{lähtevä} \quad (32)$$

(Lyytikäinen & Mäkelä 2013)

Koska säiliö on pystyseinäinen, saadaan sen korkeus hajottamalla osoittaja pienemmiksi termeiksi eli pohjan pinta-alaksi sekä korkeudeksi, ajalle ei tapahdu tässä kohtaa mitään, koska yhä edelleen kyseessä on tilavuus tiettyä ajan jaksoa kohti:

$$\frac{d(Ah)}{dt} = f_{tuleva} - f_{lähtevä} \quad (33)$$

A säiliöpohjan pinta-ala

h säiliönkorkeus. (Lyytikäinen & Mäkelä 2013)

Seuraavaksi siirrämme pohjan pinta-ala termi pois osoittajasta murtoluvun eteen.

$$A \frac{dh}{dt} = f_{tuleva} - f_{lähtevä} \quad (34)$$

(Lyytikäinen & Mäkelä 2013)

Lopuksi jaetaan pohjan pinta-ala pois. Ja näin ollen saamme yhtälöksi pinnankorkeuden ajansuhteen, mutta voimme ilmoittaa yhtälön pelkällä pinnankorkeuden merkinällä  $h$ .

$$\dot{h} = \frac{1}{A} \cdot (f_{tuleva} - f_{lähtevä}) \quad (35)$$

(Lyytikäinen & Mäkelä 2013)

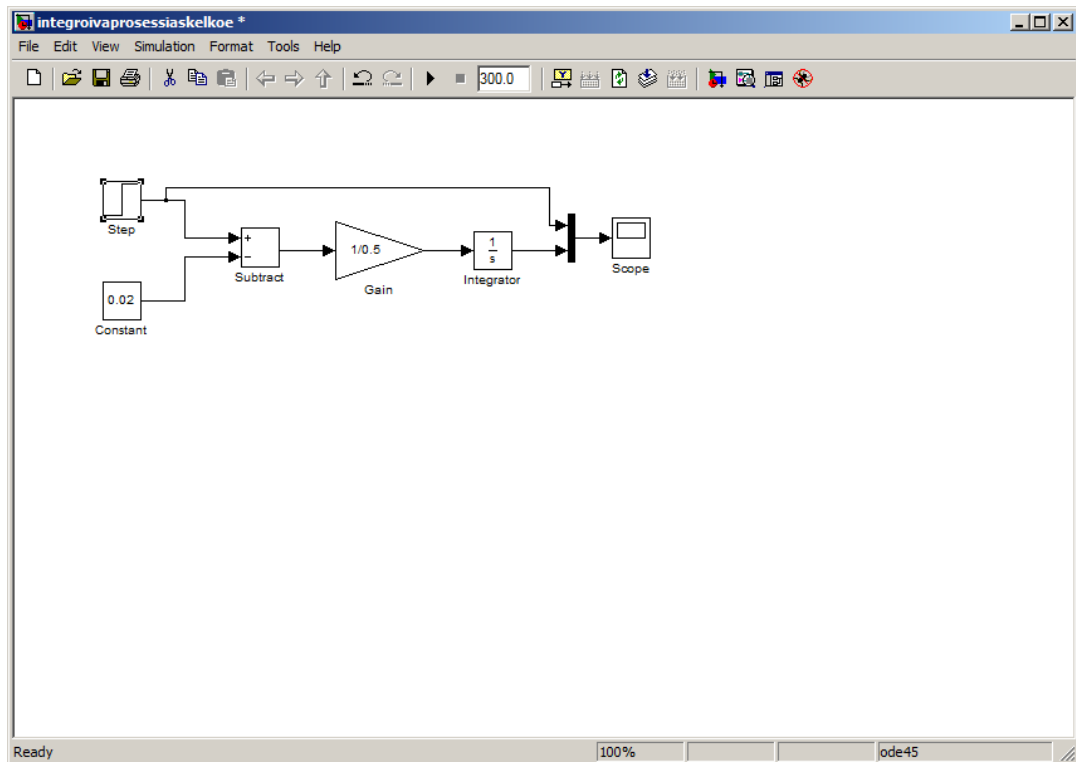
Kääntämällä yhtälö toisin päin voimme aloittaa Simulink mallin luomisen.

$$(f_{tuleva} - f_{lähtevä}) \cdot \frac{1}{A} = \dot{h} \quad (36)$$

(Lyytikäinen & Mäkelä 2013)

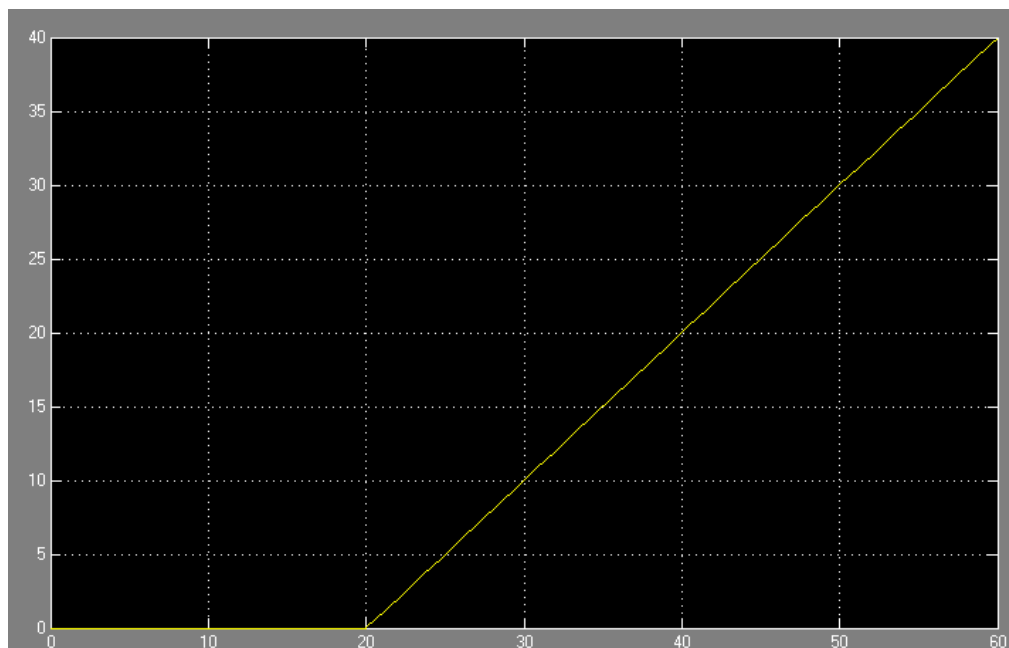
Meitä kiinnostaa nähdä pinnankorkeus joten merkitsemme sen Scope-funktiolla jotta voimme tutkia pinnankorkeuden dynamiikkaa. Koska pinnankorkeus on derivaatta, tehdään vastatoimitus integraattorilla. Vakio  $1/A$  saadaan Gain-funktiolla, joka suorittaa kertolaskun Sisälle annetulla arvolla (tässä tapauksessa  $1/0.01$ ). Viimeisenä toimenpiteenä suoritamme vähennyt laskun summaimen avulla. Tulo- ja lähtövirtauksen arvoja muutetaan vakio kertoimien avulla. Kuvassa 22 on valmis Simulink-ohjelma, johon olen asettanut valmiiksi  $0.03 \text{ m}^3/\text{s}$  ja  $0.02 \text{ m}^3/\text{s}$ . Emme ota huomioon lämpötilaa ja siitä seuraavaa tiheyden muutosta (koska vaikutus on minimaalinen) joten voidaan myös ilmaista, että säiliöön virtaa nestettä  $30 \text{ l/s}$  ja säiliöstä lähtee  $20 \text{ l/s}$ .





Kuva 22. Prosessin Simulink-ohjelma

Malli täytyy rakentamisen jälkeen testata. Tehdään askelkoe, jossa tulovirtaus on aluksi  $0.02 \text{ m}^3/\text{s}$ . 20 sekuntia myöhemmin nostamme tulovirtauksen arvoon  $0.03 \text{ m}^3/\text{s}$ .



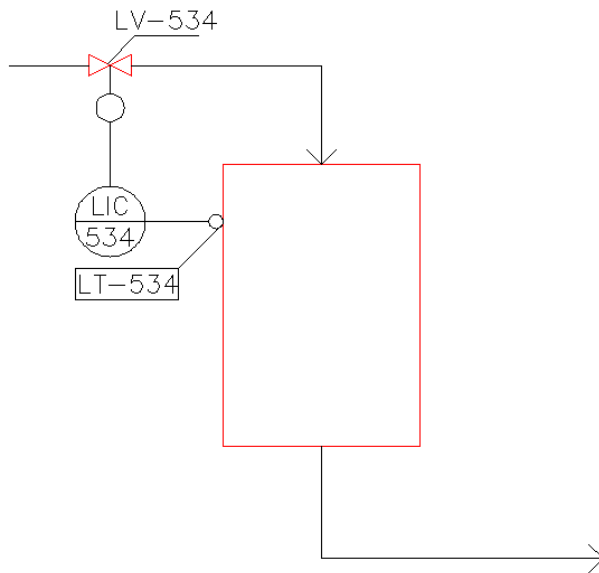
Kuva 23. Prosessin simuloitu askelvaste

Kuvan 23 trendikäyrästä voimme varmistua, että kyseessä on integroivan prosessin trendikäyrä. Trendikäyrästä voimme tulkita keltaisen viivan, joka edustaa prosessin pinnankorkeuden muutosta, että tekemämme minimaalinen muutos saa pinnankorkeuden nousemaan todella nopeasti. Y-akselilla esitetty pinnankorkeus metreinä kertoo, että pinnan korkeus kasvaa 1 m/s nopeudella. Lisäksi säiliön pieni koko varmistaa sen, että pinnankorkeus nousee holtittomasti ja nopeasti.

Trendikäyrän perusteella integroivalla prosessilla ei ole ylä- tai alarajaa vaan sen rajat täytyy luoda rakenteellisesti (Säätö- ja ohjaustekniikka 2005). Kuten huomaamme kuvasta pinta nousisi 40 m eli todellisuudessa säiliön pinta olisi vuotanut yli (avosäiliö) tai pysynyt kokoajan aivan täynnä (suljettu säiliö). Integroivan prosessin tasapainotila on, kun tulo- ja lähtövirtaus ovat yhtä suuria. Jos teemme kokeen toisin päin tilanne päättyisi siihen, että säiliö olisi tyhjä ja ulospumppaava pumppu hajoaisi tai vaurioituisi. Prosessia voitaisiin parantaa asentamalla tulo- ja lähtövirtaukseen omat säätimet jolloin pinnankorkeutta voidaan hallita. Toinen mahdollisuus olisi pintakytkin ja pumppun vaihtaminen magneettiventtiileiksi.

### 6.1.3 Läpivirtaussäiliön pintasäätö

Tässä esimerkissä on kuvan 24 mukainen läpivirtaussäiliö. Säiliön poistoputkessa ei ole estettä, vaan virtaus on vapaa. Säiliöön virtaavan veden määrää hallitaan venttiilillä. Läpivirtaussäiliötä voitaisiin käyttää esimerkiksi keinotekoisen viiveen luomisessa, koska säiliö tuo mukanaan yhden aikavakion.



Kuva 24. Läpivirtaussäiliöprosessin PI-kaavio

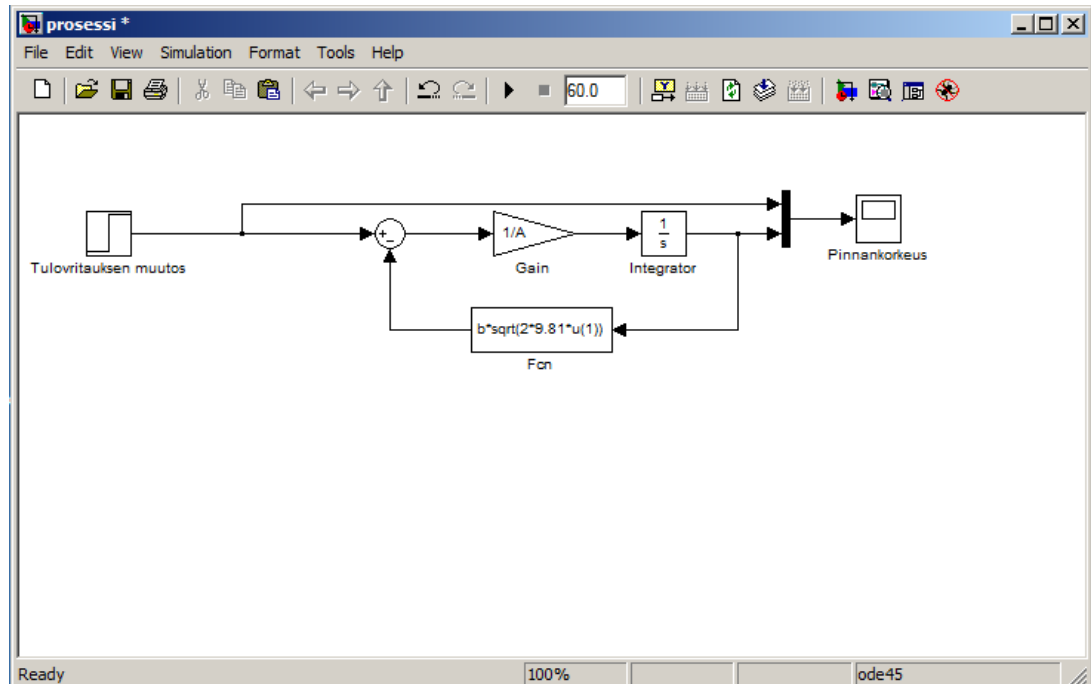
Läpivirtaussäiliön mallintamiseksi meillä on käytössämme kaksi kaavaa. Säiliön pinnankorkeuden yhtälö on tässä prosessissa sama, kuin kappaleessa 6.1.1 esiintyvä kaava 36. Säiliön poistoputkessa ei ole estettä veden virtaukselle, joten joudumme soveltamaan tässä Bernoullin lakia, joka esiintyy kaavassa 37. Yhdistämällä kaava 36 kaavaan 37 voidaan rakentaa kuvan 25 mukainen Simulink-ohjelma. Suurin muutos kappaleessa 6.1.1 esiintyneeseen integroivan säiliön malliin on, että poistoputken virtaus on omassa lohkoissaan, koska sitä ei voi kirjoittaa Simulinkin komponenttien avulla. Lohkossa  $u(1)$  on pinnankorkeus, mutta sitä ei ole voitu määritellä koska se on muuttuva.

$$f_{\text{lähtevä}} = b\sqrt{2gh} \quad (37)$$

$b$  poistoputken pinta-ala

$g$  putoamiskiihtyvyys

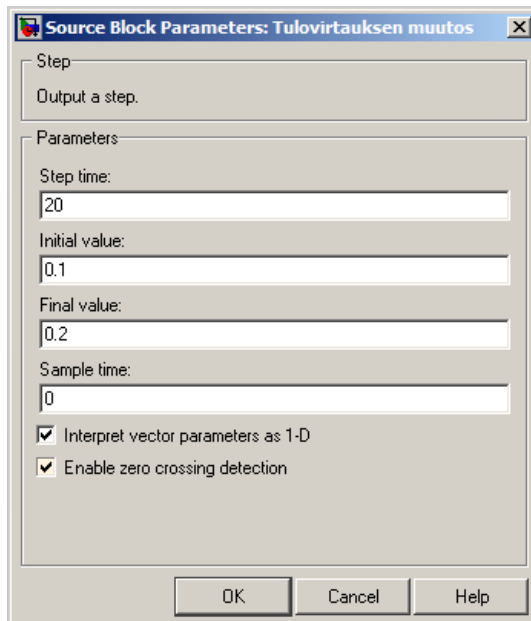
h veden pinnankorkeus säiliössä. (Lyytikäinen & Mäkelä 2013)



Kuva 25. Vapaavirtaus prosessin Simulink-ohjelma

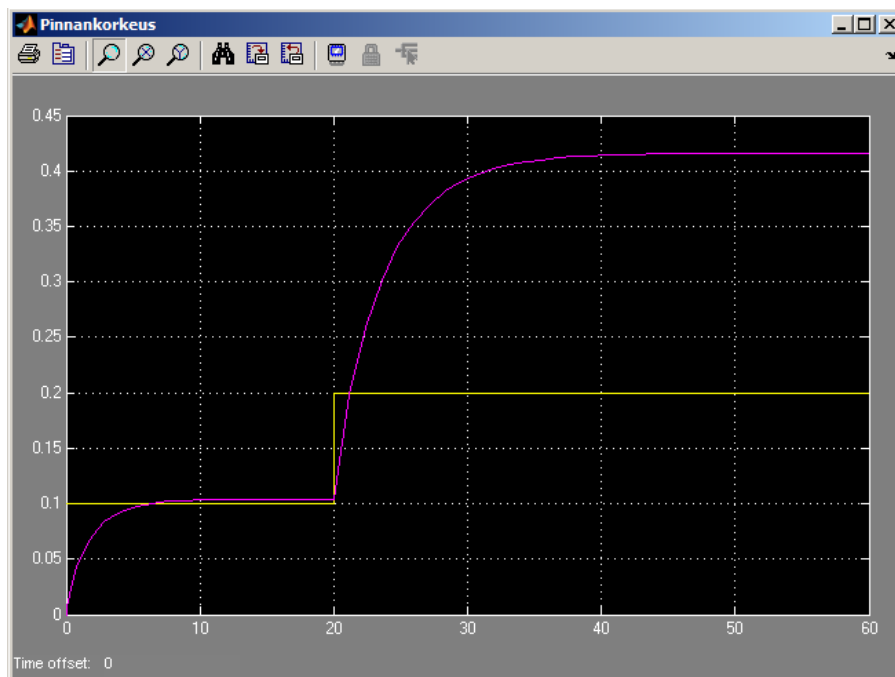
Laitetaan pohjan pinta-alan parametreiksi  $1 \text{ m}^2$  ja ulostulo putkiston pinta-alaksi  $0.07 \text{ m}^2$ , joka on halkaisijaltaan 30 cm. Annoin kyseiset arvot komentoina Matlab-ohjelman komentoriville, koska ohjelma lukee arvoja myös sieltä. Tein testejä muutamilla eri pinta-aloilla ja 0.07 toimi parhaiten. Esimerkiksi halkaisijaltaan 15 cm pienempi ulostuloputki, jolloin vedenpinta nousi todella nopeasti korkealle.

Tehdään askelkoe, jossa laitetaan ensimmäiseksi askeleen aika, joka tapahtuu 20 sekuntia simuloinnin käynnistämisen jälkeen. Tämän jälkeen asetetaan virtaukset ja annetaan alkuarvoksi  $0.1 \text{ m}^3/\text{s}$ , joka muuttuu  $0.2 \text{ m}^3/\text{s}$ . Aikojen ja muutoksen suuruuden määrittäminen Step-funktiossa näkyy kuvassa 26. Simulointiajaksi asetetaan 60 sekuntia.



Kuva 26. Simulink-ohjelman Step-funktion parametroitintia

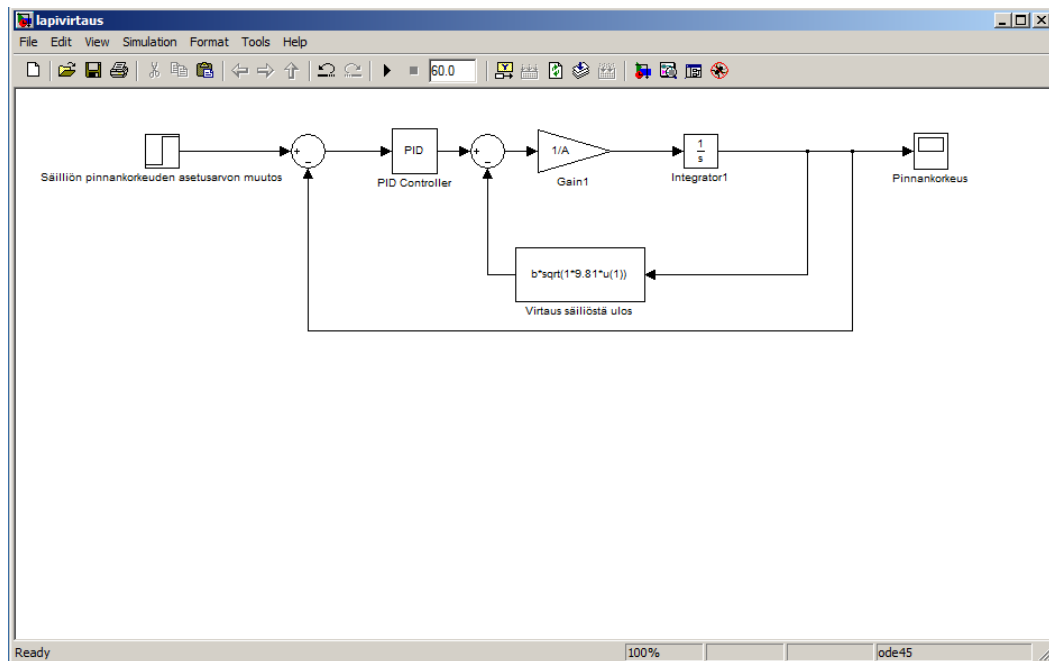
Simuloinnin tuloksena saadaan kuvan 27 mukainen trendikäyrä jossa keltainen viiva on prosessiin tehty ohjausmuutos ja violetti viiva on pinnankorkeuden muutos. Se kertoo meille, että prosessi on ensimmäisen kertaluvun ja viiveen -systemi. Viivettä prosessissa ei ole, mutta laittamalla esimerkiksi ”Transport delay”-lohkon, voitaisiin aiheuttaa prosessille viivettä. 20 sekunnin kohdalla näkyy asettamani viivytetty askel.



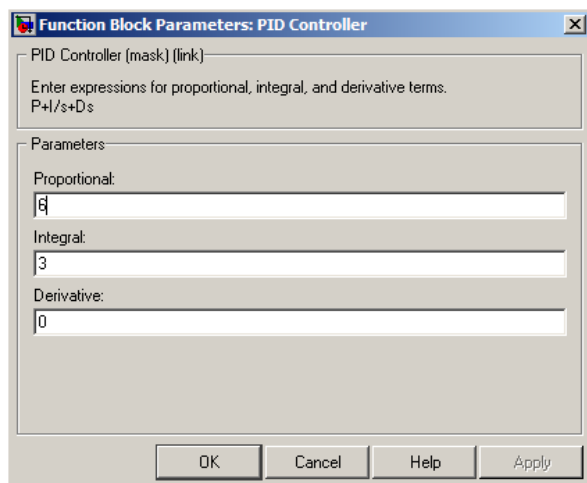
Kuva 27. Prosessille tehty askelvastekoe simuloimalla.

PID-säädin lisätään prosessin alkuun ennen summausta. Tämän jälkeen PID-säätimelle laitetaan sen vaatima oma ero suure, joka on pinnankorkeuden asetusarvon ja pinnan korkeuden erotus. Kuvassa 28 on läpivirtaussäiliön Simulink-ohjelma PID-säätimellä varustettuna. Parametroin säätimen kokeilun avulla ja tulon siihen tulokseen, että paras säädön tulos tulee, kun prosessin vahvistus on 6 ja integrointiaika on 3. Kuvassa 30 ja 31 trendikäyrä, jossa keltainen viiva on prosessiin tehty ohjausmuutos ja violetti viiva on pinnankorkeuden muutos, voidaan nähdä millainen ero säätimessä on, kun säädin ei ole viritetty ja se on viritetty. Kuvassa 30 on tulos, kun säädintä ei ole viritetty ja kuvassa 31 säädin on viritetty vahvistusarvolla 6 ja integrointi ajalla 6.

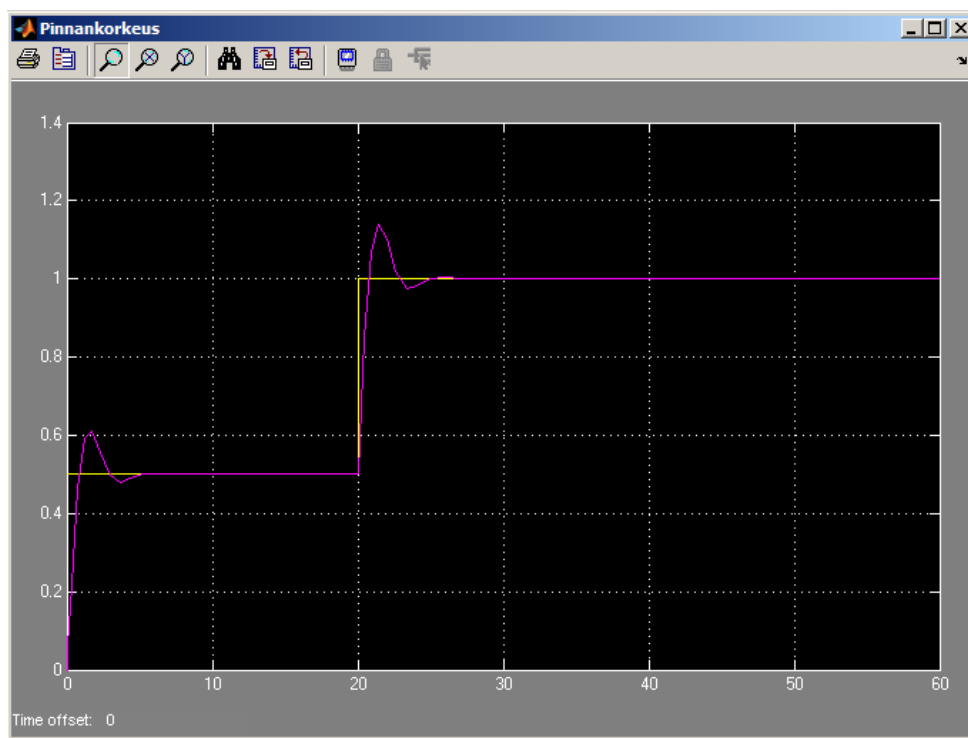
Voidaan todeta, että säätöpiirin virittämisellä voidaan päästä hyviin lopputuloksiin. Vaikka tässä ei käytetty aikaisemmin mainittuja viritystapoja, voimme nähdä simuloimisen tuoman edun. Simuloinnissa pystyin nopeasti lyhyen ajan sisällä kokeilemaan erilaisia parametreja. Esimerkiksi aidon prosessin kanssa olisi mennyt huomattavasti pidempi aika. Kappaleessa 6.2.2 havainnollistetaan paremmin säätöpiirin viritystä lambda-virityksen avulla.



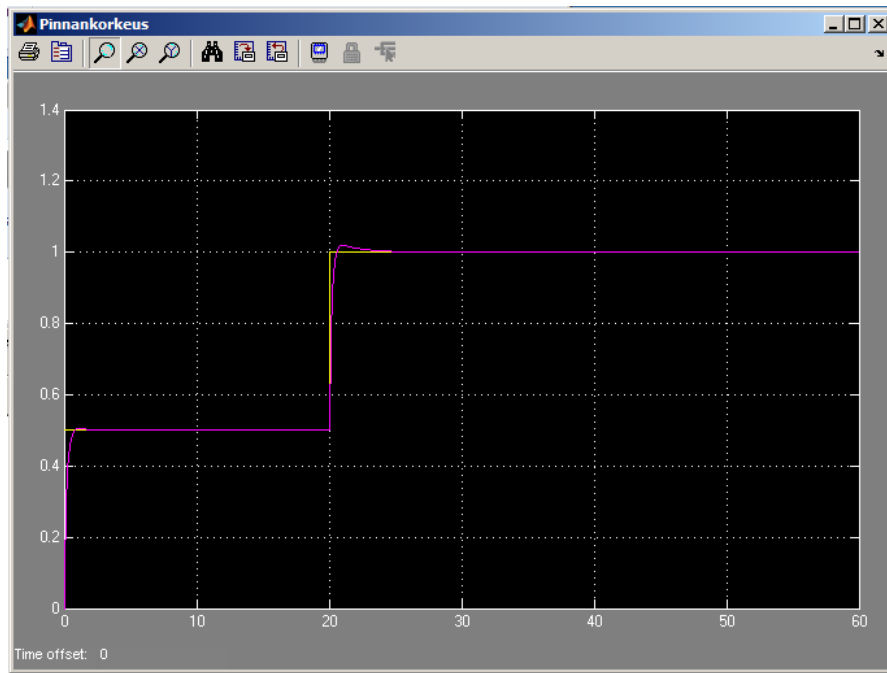
Kuva 28. Läpivirtaussäiliön Simulink-ohjelma säätimellä varustettuna



Kuva 29. PID-säätimen asetusarvot Matlabissa



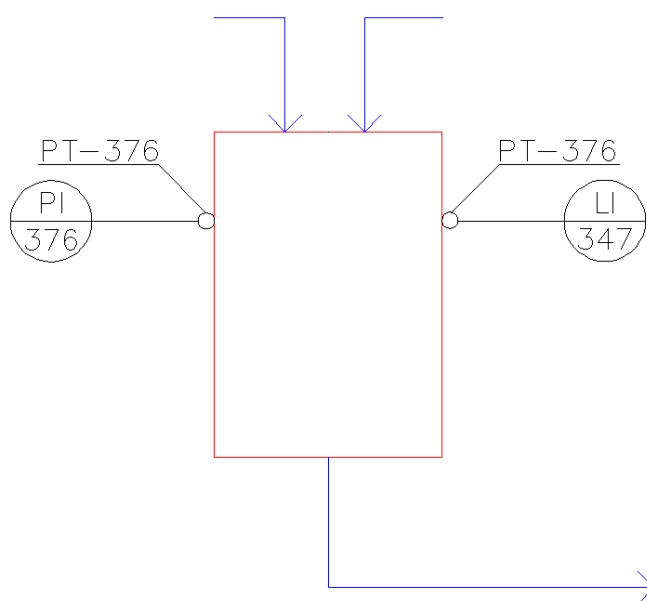
Kuva 30. Simuloitu askelvaste ilman parametrien muuttamista



Kuva 31. Simuloitu askelvaste parametrien sovituksen jälkeen

#### 6.1.4 Läpivirtaussäiliön lämpökapasiteetti

Kuvassa 32 on läpivirtaus säiliön lämpökapasiteettiprosessi. Prosessissa säiliöön tulee sekä kylmää että kuumaa vettä ja säiliöstä poistuva vesi on haluamamme lämpöistä. Prosessissa kylmän ja kuuman veden virtauksien suhde vaikuttaa pinnankorkeuteen ja veden lämpötilaan, jolloin kyseessä on monimuuttujamalli.



Kuva 32. Läpivirtaussäiliön lämpökapasiteettiprosessi



Prosessin pinnankorkeuden määrää pohjan pinta-ala, kuuman ja kylmän veden määrä, sekä poistuvan veden määrä. Pinnankorkeuden tasapainotila on, kun vettä tulee yhtä paljon, kuin sitä poistuu. Lämpötilan määrää kuuman ja kylmän veden suhde. Esimerkiksi jos halutaan vaikkapa 40 asteista vettä, vähennetään kuuman veden määrää ja lisätään kuuman veden määrää. Kaavassa 38 on prosessin pinnankorkeuden määrittävä yhtälö. Veden virtaus pois säiliöstä on toteutettu vapaalla virtauksella, jolloin yhtälöön pätee Bernoullin laki, joka löytyy kaavasta 39. Kaavassa 40 otetaan huomioon lämpöenergia, joka siirtyy veteen. Lämpöenergiaa vaikuttaa lämpötilojen erotus, aineen massa ja ominaislämpökapasiteetti. Lämpöenergian muutos on esitetty kaavassa 41 ja, kun yhtälöä hieman pyöritetään, päästään kaavan 42 muotoiseen lopulliseen kaavaan veden lämpötilan muutoksesta.

$$\frac{dV}{dt} = \frac{adh}{dt} = f_{tuleva} - f_{lähtevä} = (f_{kuuma} + f_{kylmä} - f_{lähtevä}) \cdot \frac{1}{A} \quad (38)$$

V	säiliön tilavuus
a	säiliön pohjan pinta-ala
h	säiliön korkeus
$f_{tuleva}$	tulevan veden tilavuusvirtaus
$f_{lähtevä}$	säiliöstä lähtevän veden tilavuusvirtaus
$f_{kuuma}$	kuuman veden tilavuusvirtaus
$f_{kylmä}$	kylmän veden tilavuusvirtaus. (Lyytikäinen & Mäkelä 2013)

$$f_{lähtevä} = b\sqrt{2gh} \quad (39)$$

b	poistoputken pinta-ala
g	putoamiskiihtyvyys

$h$  säiliön koko. (Lyytikäinen & Mäkelä 2013)

$$Q = cm\Delta T \quad (40)$$

$Q$  lämpömäärä

$c$  aineen ominaislämpökapasiteetti

$m$  aineen massa

$\Delta T$  lämpötilaero. (Lyytikäinen & Mäkelä 2013)

$$ahdc \frac{dT}{dt} = f_{kuuma}dc(T_{kuuma} - T) - f_{kylmä}dc(T - T_{kylmä}) \quad (41)$$

$d$  aineen tiheys

$T$  säiliössä olevan veden lämpötila

$T_{kuuma}$  kuuman veden lämpötila

$T_{kylmä}$  kylmän veden lämpötila. (Lyytikäinen & Mäkelä 2013)

$$\frac{dT}{dt} = \frac{(f_{kuuma}(T_{kuuma} - T) - f_{kylmä}(T - T_{kylmä}))}{ha} \quad (42)$$

Yhtälöistä voidaan päätellä, että prosessin tasapainotila on, kun kuumaa ja kylmää vettä tulee säiliöön yhtä paljon, kuin sitä poistuu poistoputkesta. Säiliössä olevan veden lämpötilan määrää kylmän ja kuuman veden virtausten suhde. Mikäli haluamme viilentää vettä, lisätään kylmän veden virtausta ja vähennetään kuuman veden tilavuusvirtausta. Tasapainotilan voi laskea määrittelemällä derivaatat 0-arvoisiksi. Näin saadaan yhtälöpari. Oletetaan, että tulevan kylmän veden lämpötila on 10 °C ja tulevan kuuman veden lämpötila on 80 °C, lisäksi oletamme säiliön pinta-alan olevan 1m<sup>2</sup> ja korkeuden olevan 1 m, poistoputken pinta-ala on 0,02 m<sup>2</sup>. Säiliön veden lämpötila on 65 °C. Näiden tietojen perusteella, voidaan laatia yhtälöpari.

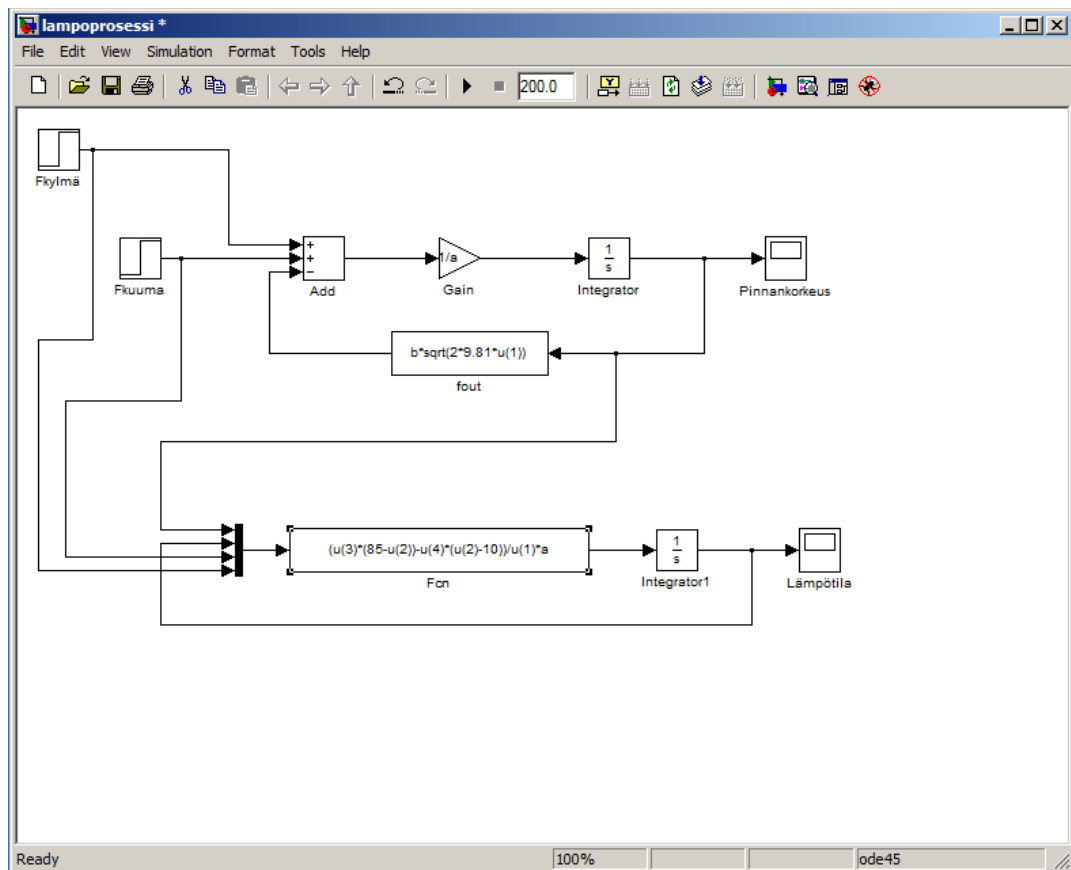
$$(f_{kuuma} + f_{kylmä} - b\sqrt{2gh}) \cdot \frac{1}{A} = 0 \quad (43)$$

(Lyytikäinen & Mäkelä 2013)

$$\frac{(f_{kuuma} \cdot (T_{kuuma} - T) - f_{kylmä} \cdot (T_{kylmä} - T))}{h \cdot A} = 0 \quad (44)$$

(Lyytikäinen & Mäkelä 2013)

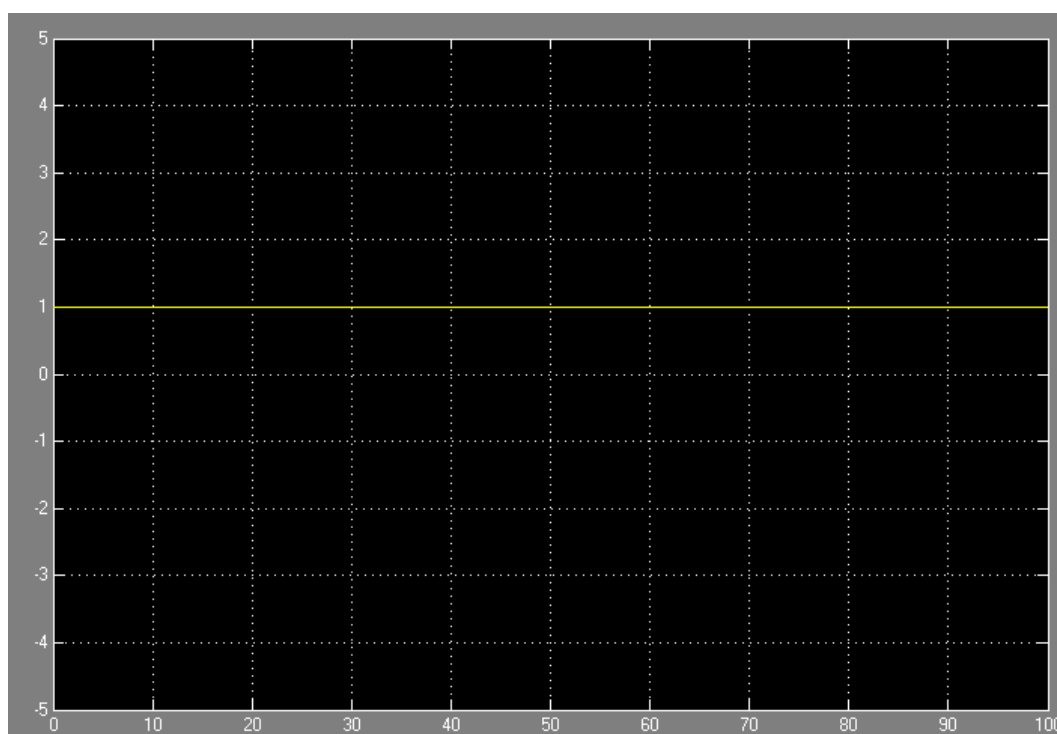
Laskelmieni mukaan säiliöön pitää tulla vettä 0.0649 m<sup>3</sup>/s ja kylmää vettä 0.0236 m<sup>3</sup>. Voimme testata ohjelman toimivuutta rakentamalla kaavojen 40 ja 44 perusteella Simulink-ohjelma, joka on kuvassa 33. Kuvan ylempi Simulink-ohjelman osa perustuu kaavaan 38 ja alempi kaavaan 42. Lämpötilat ja virtausmäärät ovat muuttuvia, joten ne laitettiin muuttujan arvoilla eli u(1) Voimme huomata kuvasta, että kyseisestä mallirakenteesta tuli melko monimutkainen. Koska kumpikin kaava vaikutti toisiinsa, jouduin käyttämään mux-funktiota, jotta sain kaikki signaalit yhteen paikkaan. Täytyy muistaa kyseisen toiminnon kanssa, että esimerkiksi alemman funktiolohkon kanssa muuttujan kerrotaan siinä järjestyksessä, kun ne tulevat muxiin. Järjestys lasketaan ylhäältä alaspäin.



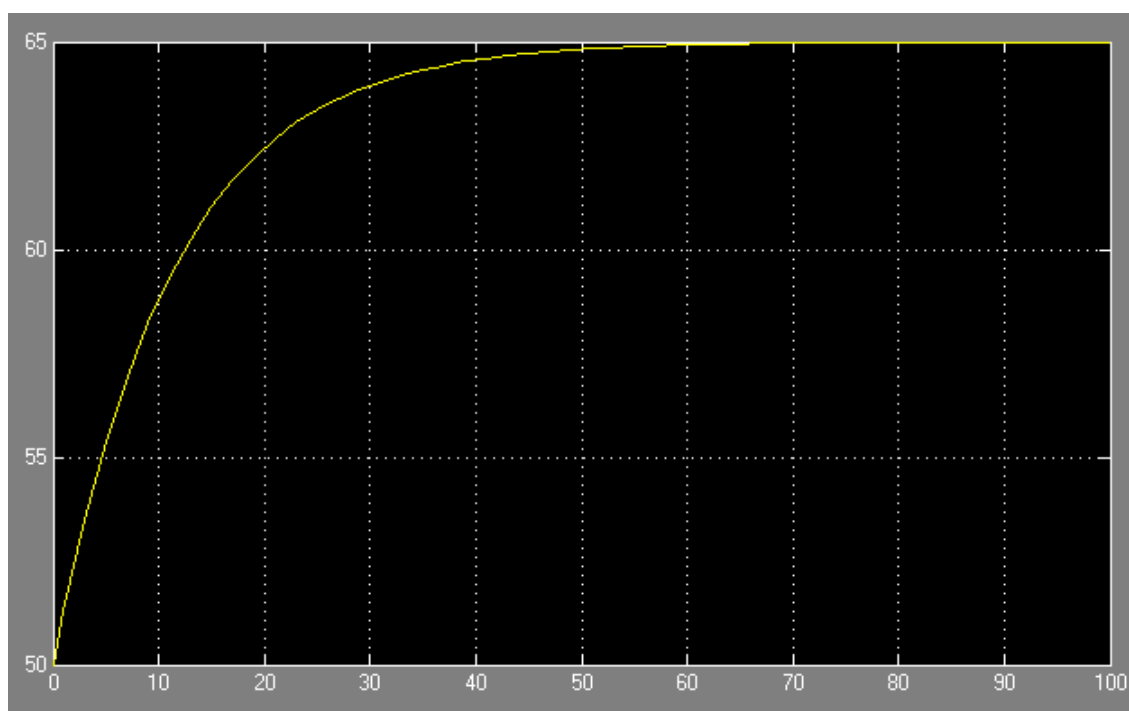
Kuva 33. Lämpöprosessin Simulink-ohjelma

Syötin Step-funktioihin tasapainotilasta saadut arvot ja painoin simulointi-painiketta. Sain tulokseksi, että laskelmani pitivät paikkansa, koska kuvissa 34 ja 35 käyttäytymisen on, kuten halusin eli pinnankorkeus on tasan metri ja lämpötila 65 °C. Tehdään prosessille vielä koe, jossa nostan kylmän veden tilavuusvirtauksen arvoon 0.025 m/s ja kuuman veden virtaus pidetään vakiona. Simulointiaikaa nostetaan 250 sekuntiin. Kuvista 36 ja 37 voimme huomata, että prosessi on tasaantuva, vaikka sinne pumpattaisiin enemmän vettä, kuin sieltä lähtee ulos, pinta tasaantuu. Samalla voimme huomata, että kylmän veden lisääminen vaikutti lämpötilaan.

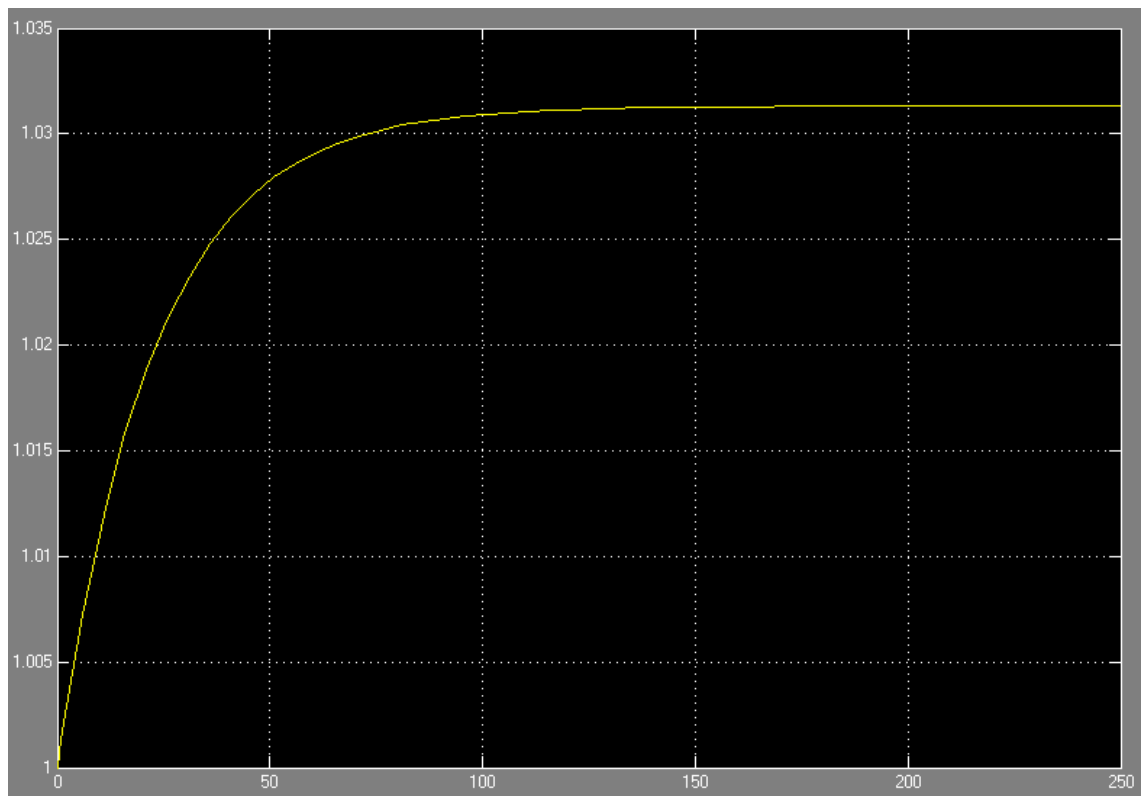
Prosessi on monimuuttujamalli, joten sen hallinta perinteisillä PID-säätimillä on vaikeaa. Tässä esimerkissä esiintynyttä monimuuttujamallia voitaisiin säätää esimerkiksi sumean logiikan avulla, jolloin pystyttäisiin paremmin huomioimaan pinnankorkeuden ja lämpötilan suhde.



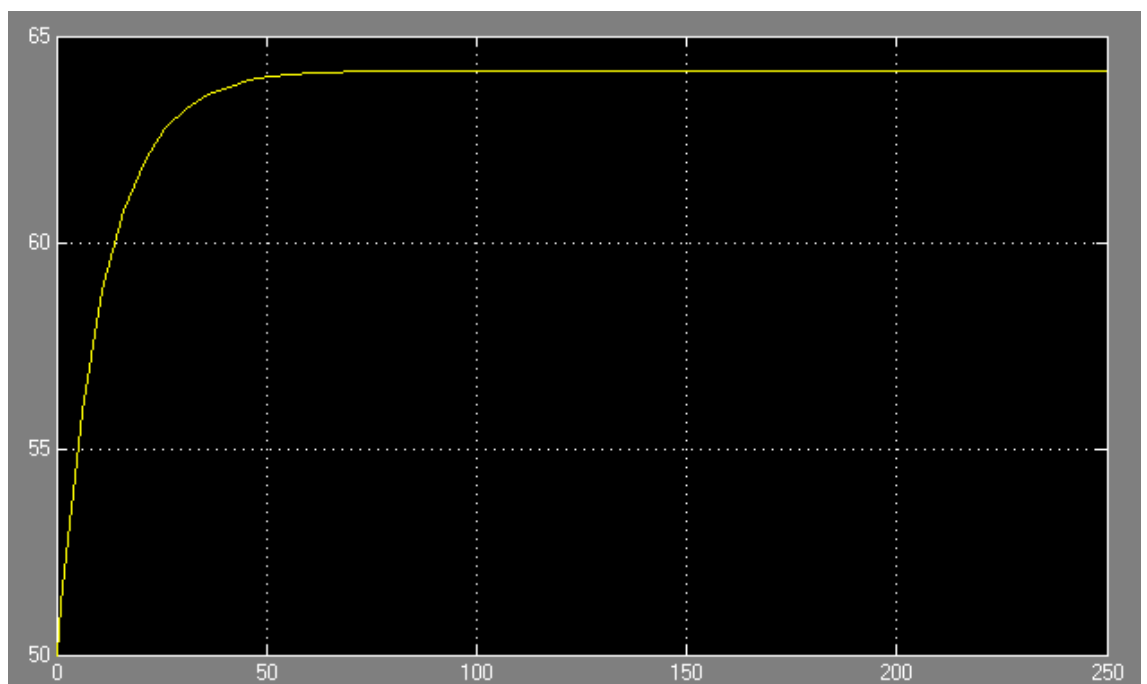
Kuva 34. Pinnankorkeuden muutos tasapainotilassa



Kuva 35. Lämpötilan muutos tasapainotilassa



Kuva 36. Pinnankorkeuden muutos lisättäessä kylmää vettä.



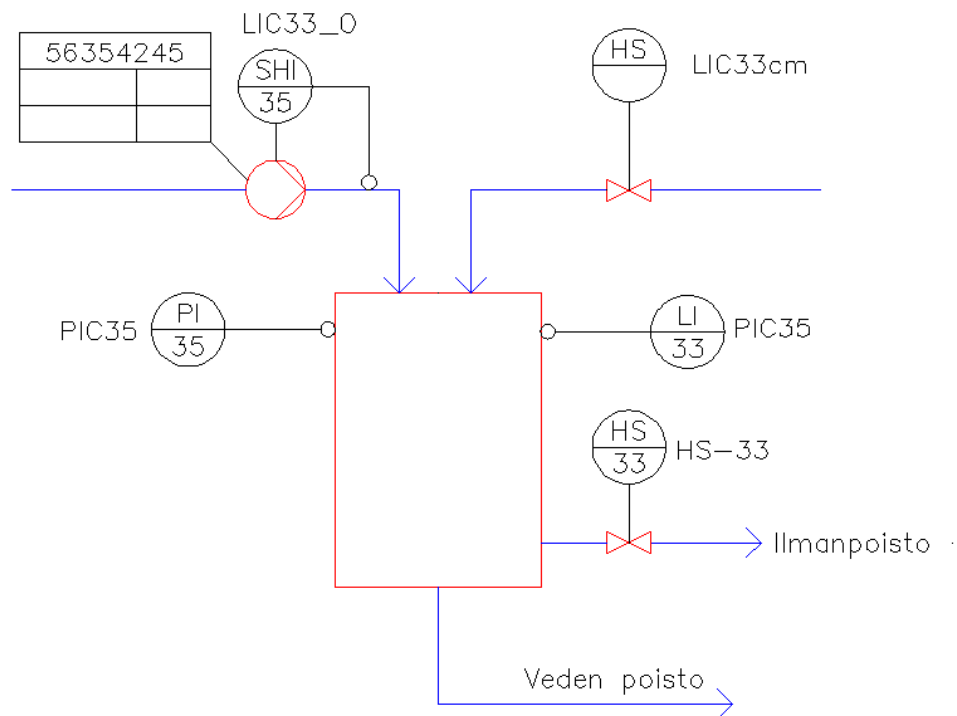
Kuva 37. Lämpötilan muutos, kun kylmää vettä on lisätty

## 6.2 Kerättyyn dataan perustuvat mallit

Olemassa olevan prosessin mallintaminen vaatii tiedonkeruujärjestelmän, joka kerää mittaustietoa prosessista. Mittaustieto voidaan ottaa, joko trendikäyränä tai aikaleimattuna taulukkona. Tämä riippuu siitä, millaiset tietotekniset työvälineet ovat käytössä. Esimerkiksi ohjelmistojen yhteensopimattomuus käyttöjärjestelmien kanssa voi aiheuttaa sen, että on helpompi suorittaa askelkoe ja sen perusteella tehdä matemaattinen malli. Toisaalta voi olla myös tilanne, että tiedonkeruu Excel-tilaukseen on mahdollista, jolloin voidaan mittaustietojen perusteella tehdä ARX-malli. Tässä kappaleessa tehdään sekä ARX-malli ja askelkoe. Lisäksi tutkitaan tilannetta, jossa askelkoe ja satunnaispulssikoe menivät pieleen ja selvitetään, mikä aiheutti ongelmat.

### 6.2.1 Painesäiliön ARX-malli

Olemme aikaisemmin keränneet dataa painesäiliöprosessista ja tiedot on siirretty Matlab-ohjelmistoon. Painesäiliöprosessin PI-kaavio on kuvassa 38. Prosessista mitataan painetta ja pinnankorkeutta. Paineeseen ja pintaan vaikuttavat ON/OFF-venttiilit, joilla säädetään syöttö- ja ulospuhallusventtiileitä. Pintaan ja paineeseen voidaan myös vaikuttaa pumpulla LIC33\_O. ON/OFF-venttiilien tietoja ei ole voitu rekisteröidä, joten niiden mallinnuksessa päälle-pois-komennot joudutaan generoimaan itse. Tässä esimerkissä keskitymme tarkastelemaan pumpun kierrosnopeuden vaikutusta säiliön paineeseen.



Kuva 38. Painesäiliön PI-kaavio

Aloitetaan lataamalla tiedosto lataamalla tiedot Matlabin komennolla "load painepintadata.mat" ja tarkistetaan mitä tietoja tiedosto pitää sisällään komennolla "whos". Kuvassa 39 näkyy kaikki tiedoston sisältämät datapaketit, joihin on tallennettu tietoa prosessista. Size-kohdasta voimme päätellä, että kaikki mittaustiedot ovat samalta ajan jaksolta, koska tiedostojen koko on sama.

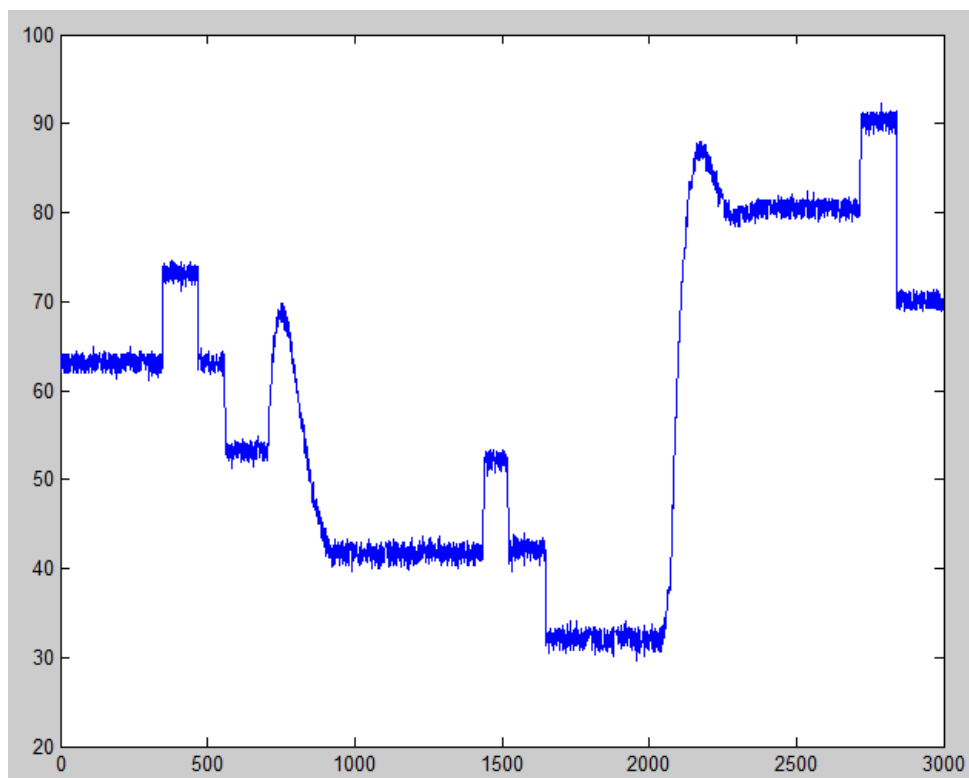


```
>> load painepintadata.mat
>> whos
```

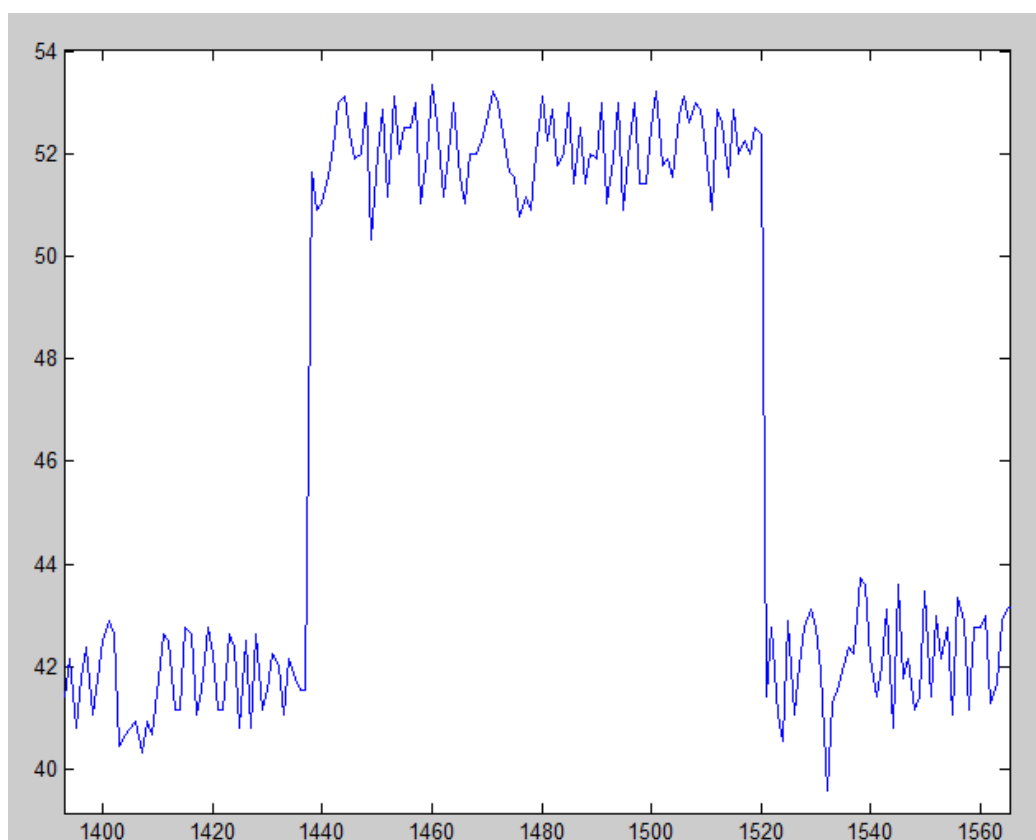
Name	Size	Bytes	Class	Attributes
FIC41	2997x1	23976	double	
FIC61	2997x1	23976	double	
FIC61_O	2997x1	23976	double	
FIC61cm	2997x1	23976	double	
LI12cm	2997x1	23976	double	
LIC33_O	2997x1	23976	double	
LIC33cm	2997x1	23976	double	
LIC42_O	2997x1	23976	double	
LIC42cm	2997x1	23976	double	
LIC51	2997x1	23976	double	
PIC35	2997x1	23976	double	
Vtot	2997x1	23976	double	
Vtotf	2997x1	23976	double	
ans	1x2	16	double	
mittaus1	2997x11	263736	double	
mittaus2	2997x11	263736	double	

Kuva 39. Tietojen lataus ja tarkistus

Lähdetään identifioimaan, kuinka syöttöpumpun kierrosnopeus vaikuttaa paineeseen. Kuvassa 39 näkyvistä tiedostoista tässä esimerkissä käytetään syöttöpumpun kierrosnopeutta, joka on nimellä LIC33\_O ja paineen mittaustietoa, joka on tallennettua nimellä PIC35. Aluksi on hyvä tarkistaa, millaista kuvaajaa tiedonkeruu on ottanut talteen. Komennolla ”plot (LIC33\_O)” aukeaa kuvan 40 näköinen 2D-kuvaaja, jossa näkyy prosessin muutokset. Kuvaajan perusteella tiedonkeruuta on suoritettu 3000 sekuntia eli 50 minuutin ajan. Lisäksi voidaan todeta, että pumpun nopeus vaihtelee todella paljon ja todella useasti. Valitsemme mielenkiintoiseksi kohdaksi 1500 sekunnin kohdalla esiintyvän nopeuden nostamisen. Tarkemman aikavälin saamiseksi käytetään Zoom-toimintoa, jossa klikkaan hiirellä kuvaajaa ja liikutan hiirtä siten, että 1500 sekunnin kohdalla oleva muutos jää laatikon sisään. Tämän jälkeen saadaan kuvan 41 mukainen kuvaaja.



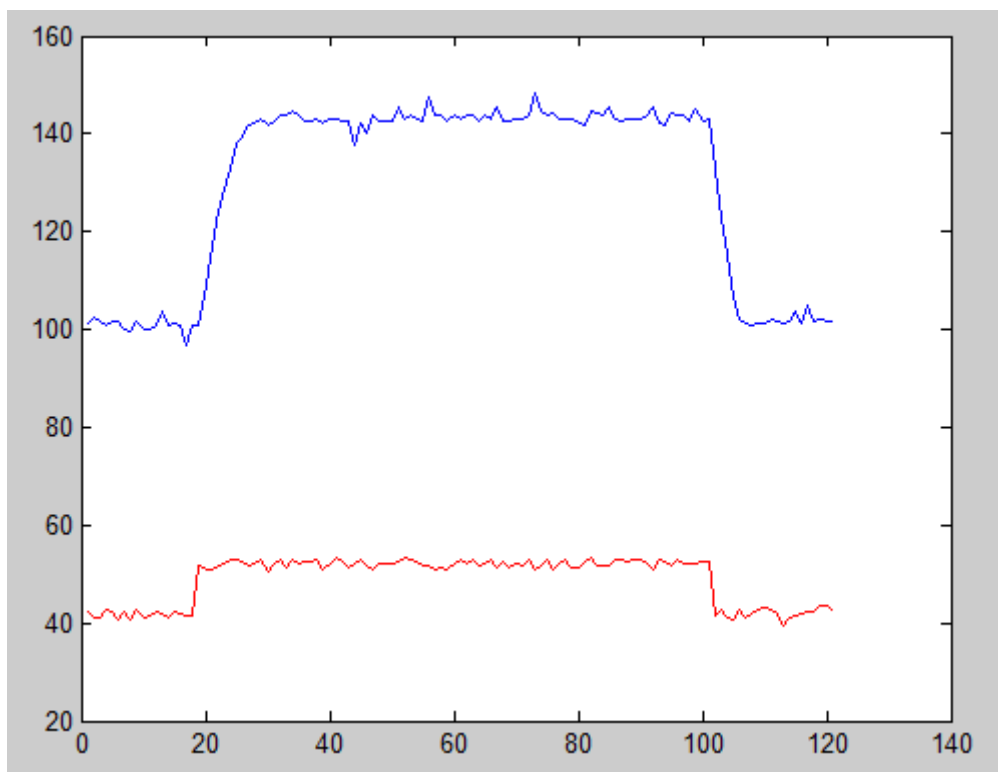
Kuva 40. Pumpun nopeuden käyttäytyminen



Kuva 41. Pumpun nopeuteen tehty askelmainen muutos

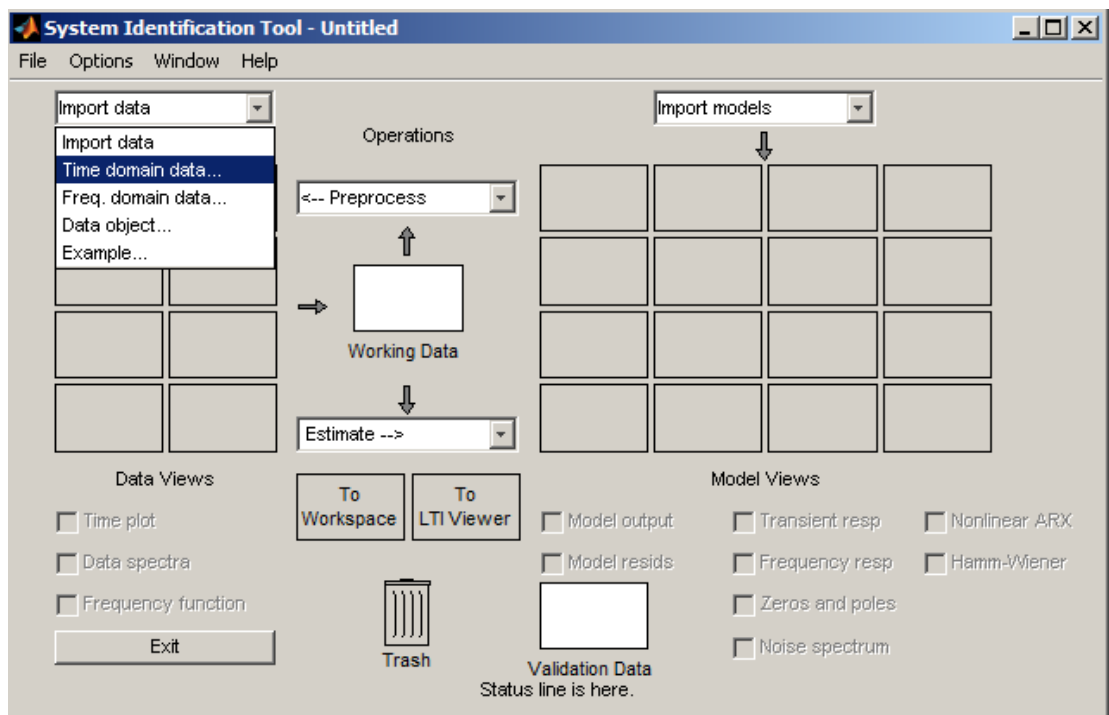
Tarkasti approksimoituna Zoom-toiminnon avulla selviää, että askel alkoi, kun aikaa oli kulunut 1437 sekuntia. Askel takaisin alas tehtiin, kun aikaa oli kulunut 1521 sekuntia. Otamme kuitenkin hieman enemmän dataa talteen ajalta jolloin prosessiin ei ole tehty muutosta. Otamme nämä ajanjaksot talteen ja verrataan sitä pinnankorkeuteen ja paineeseen, koska tuolla ajanjaksolla on täytynyt tapahtua muutoksia prosessissa. Ottaessamme tietoja ylös, muutamme samalla tietojen nimikkeet siten, että Matlab ymmärtää, mistä suureesta puhutaan eli kirjoitamme komento riville: `>>%Pick up, >> speed=LIC33_O(1420:1540);`. Samalla komennolla otetaan talteen myös paineen tiedot eli kirjoitetaan komento riville: `>>%Pick up, >> pressure=PIC35(1420:1540);`. Tämän jälkeen voidaan katsoa plot-toiminnolla kuvaajana, mitä valitsemalla ajanjaksolla tapahtui.

Paineen muutos ei välttämättä näy niin selvästi, joten laitan komentoon, että paineen arvo kerrotaan 100, jotta saadaan paineen muutos paremmin näkyviin. Kyseisellä toiminnolla ei vaikuteta paineen mittaustietoihin, vaan ainoastaan siihen, miten plot-toiminto esittää suureen meille. Kirjoitetaan komento riville seuraavan lainen komento: `plot([1:250],speed,'r',[1:250],pressure*100,'b')` jolloin saadaan kuvan 42 mukainen kuvaaja.



Kuva 42. Pumpun kierrosnopeuden ja paineen muutoksen välinen suhde

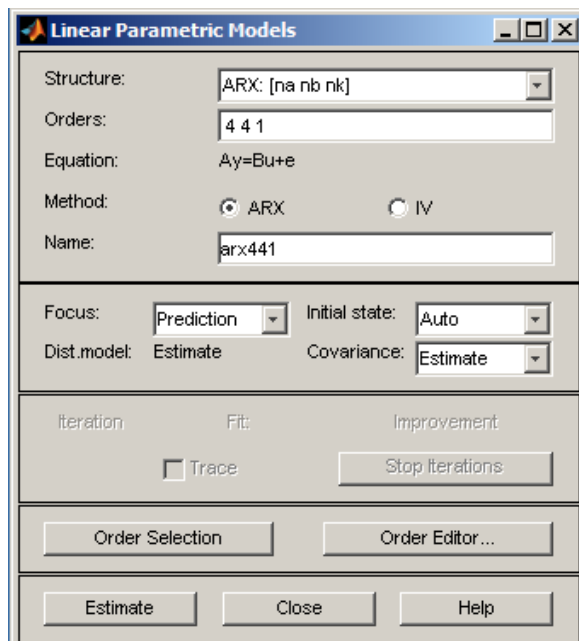
Oletimme aikaisemmin, että pumpun nopeus muutos kohdassa 1437, vaikuttaa paineeseen. Kuvan 42 perusteella voimme todeta, että oletus pitää paikkansa. Kun tietojen paikkansa pitävyys on varmistettu, voidaan tiedot ajaa Ident-käyttöliittymään ja tehdä ARX-malli. Ident-käyttöliittymään saa auki kirjoittamalla komentoriville ”ident”. Tiedot saadaan ajettua käyttöliittymään kuvassa 43 näkyvällä import-data toiminnolla, josta valitaan ”Time domain data”. Tämä valitaan, koska kyseessä on aikaleimattu dataa. Valittuamme ”Time domain data” tulee kuvan 44 mukainen ikkuna, johon tulosuureeksi (input) kirjoitetaan speed eli syöttöpumpun nopeus ja lähtöön (output) kirjoitetaan pressure. Tiedot saadaan Ident-käyttöliittymään klikkaamalla import-painiketta.



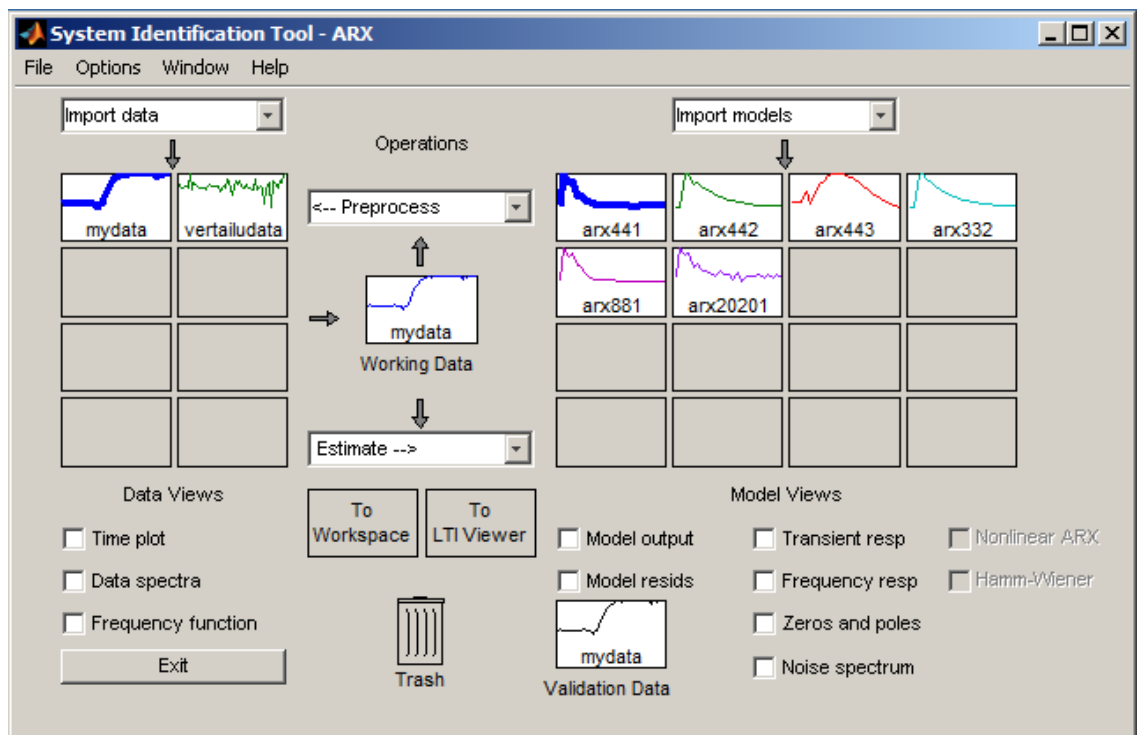
Kuva 43. Ident-käyttöliittymä

Kuva 44. Ident-käyttöliittymän tietojen tuominen

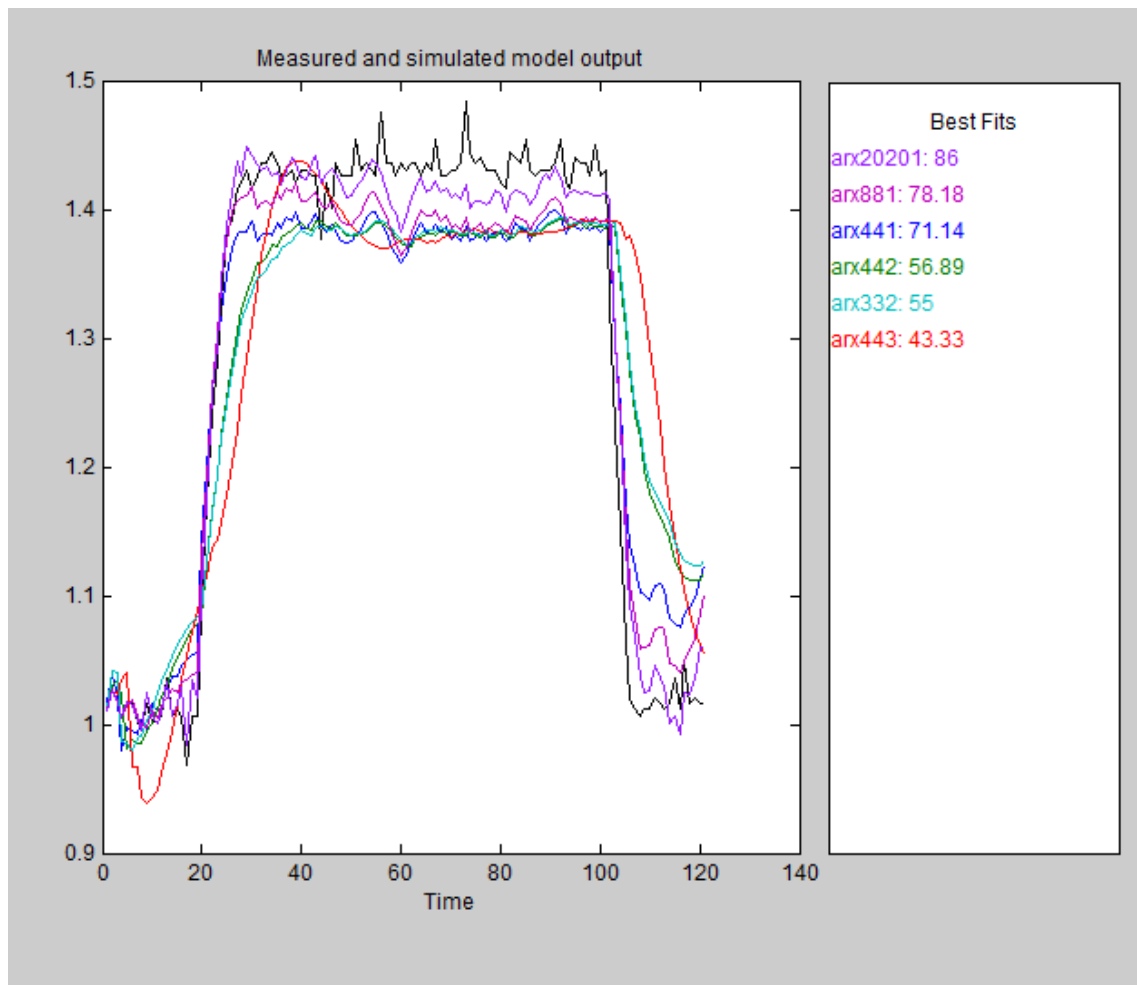
Ident-malli saadaan klikkaamalla ”Estimate” ja valitsemalla ”Linear Parametric Models”. Kyseinen valikko on kuvassa 45. Ident arvioi ARX-mallin vastaavuuden vertaamalla mitattua ulostuloa (painetta), simuloituun tulokseen. Simuloidun tuloksen tarkkuus riippuu kertoimista, jotka ARX-mallille tehdään Tein ensimmäisen ARX-mallin laittamalla parametreiksi 3,3 ja 2. Tulokseksi sain 55 % vastaavan mallin. 55 % ei ole kelvollinen prosentti määrä, vaan meidän täytyy saada mallia tarkemmaksi. Yksi tapa on viilailla kertoimia. Kokeilin eri yhdistelmiä ja minulle selvisi, että epätarkkuutta aiheutti kolmannen termin muuttaminen. Esimerkiksi 4, 4 ja 3 yhdistelmällä tuli vain 43.33 % vastaavuus, kun laitoin kolmannen termin tilalle 1, prosessin vastaavuus oli 71.14 %. Kyseisellä prosentti määrällä aletaan olla jo hyvin prosessia kuvaavassa mallissa.



Kuva 45. ARX-mallin parametointi



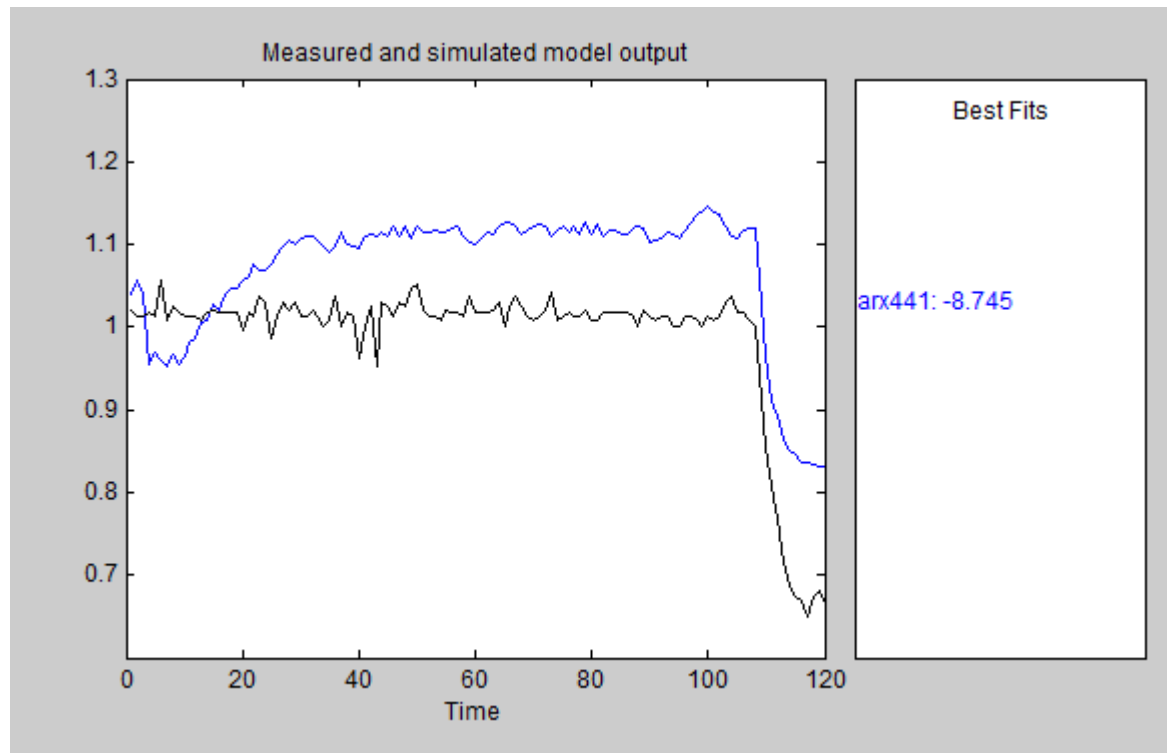
Kuva 46. ARX-käyttöliittymä



Kuva 47. Kertoimien vaikutus prosessin mallin hyvyyteen

Valitsin arx441:en testattavaksi Simulink-ohjelmaa varten, mutta varmistetaan millainen on datan vastaavuus, kun otetaan mukaan vertailudata, joka on otettu 120 sekuntia ensimmäisen näytteen jälkeen. Komennoilla `">>%Pick up, >> pressure-ref=PIC35(1541:1660);"` ja `">>%Pick up, >> speedref=LIC33_O(1541:1660);"`, saadaan otettua tiedot ylös tietyille nimille. Vertailudata siirrettään ident-käyttöliittymään samalla tavalla, kuin aikaisemmin valittu datanäyte kuvassa 44. Uusi data raahataan "Validation data" -lohkoon ja klikataan "Model Output" -ruksia, jolloin nähdään, miten hyvin data vastaa laskettua lähtöä. Sain tulokseksi -8 %. Tämä tarkoittaa sitä, että malli ei välttämättä päde kokonaisuudessaan. Vertailudatan sopimattomuuteen on vaikea löytää syytä. Ongelma saattoi tulla tietokoneesta tai datasta. Tämä on vaikea todeta, mutta tein kaikki uudelleen ja samalla tuloksella. Lähinnä ohjelmistovirhe tuli ensimmäisenä mieleen, koska tietokone temppuili ja kaatoi ohjelman. Kuitenkin toistettuani ARX-mallinnuksen uudelleen, päädyin samaan lopputu-

lokseen. Lähdön ja vertailudatan eroa saa pienennettyä nostamalla kertoimien arvoa. Kokeillaan seuraavaksi miten iso vaikutus kertoimilla on Simulink-ohjelmassa.

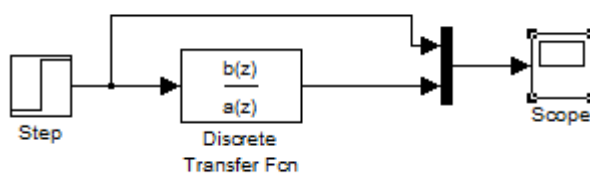


Kuva 48. Vertailudatan hyvyyden arviointi

Malli kannattaa identifiointin jälkeen testata Simulink-ohjelmassa. Raahataan työni-  
meltä arx441 workspace nimiseen lohkokon, joka näkyy kuvassa 46 keskellä. Tämän  
jälkeen haetaan kertoimet kirjoittamalla Matlabin komentoriville:

`[a,b]=arxdata(arx441)`. Avataan Simulink ja tehdään kuvan 49 näköinen lohkokkaavio.

Lohkokkaaviossa a ja b tulevat kertoimista, jotka Matlab antoi meille aikaisemmalla  
komennolla. Simulink-ohjelmistolla sain testattua painesäiliön mallin. Tuloksena si-  
muloinnista on kuvan 50 trendikäyrä, jossa keltainen viiva on askel ja violetti viiva  
paineen muutos.



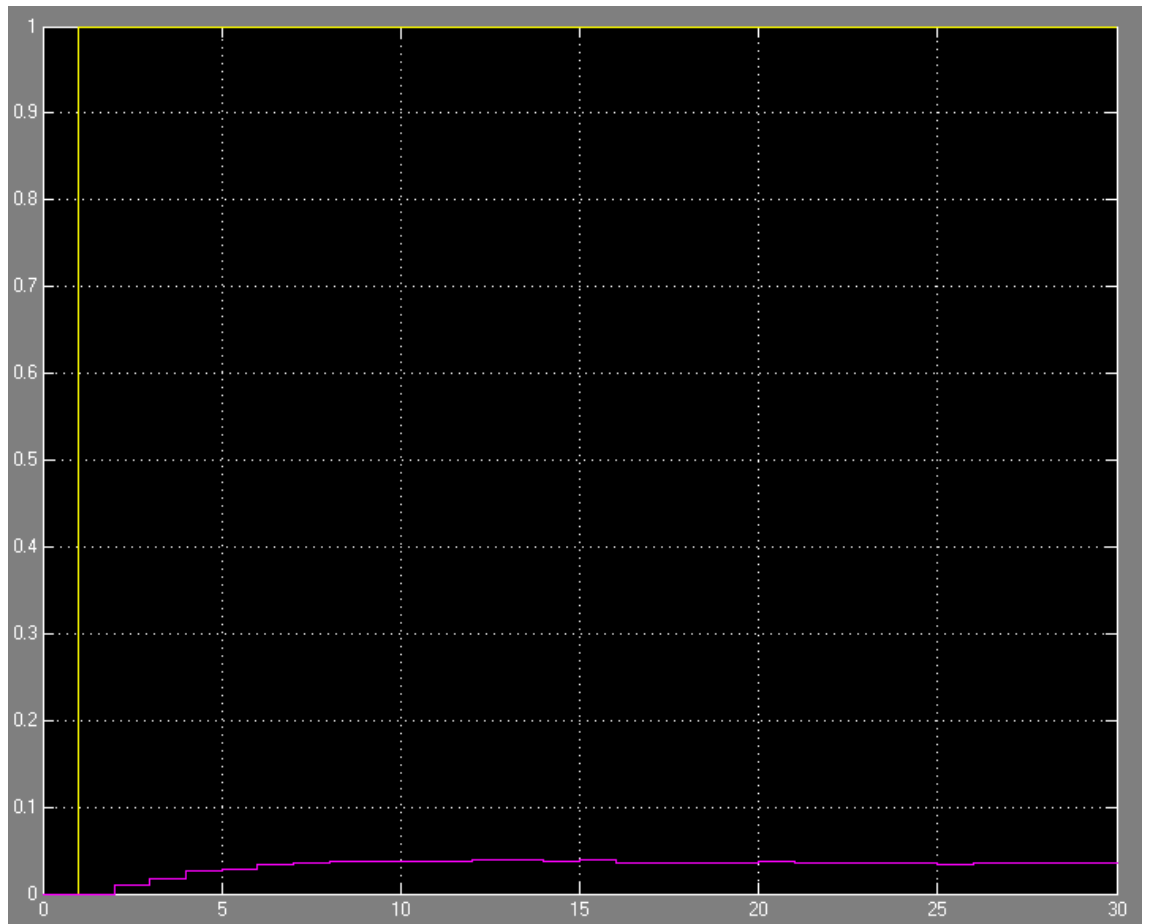
Kuva 49. Painesäiliön Simulink-ohjelma





Kuva 50. Painesäiliön trendikäyrä, alempi käyrä paine, ylempi käyrä

Aikaisempien toimenpiteiden avulla hain kertoimet, jolloin Simulink-ohjelma tunnisti uudet kertoimet ja sain kuvan 51 mukaisen trendikäyrän. Kuvasta 51 voidaan nähdä, että kertoimella on jonkin verran vaikutusta tarkkuuteen, mutta epätarkkuuksia ei voida laittaa kokonaan kertoimien syyksi. Aikaisemmin mainittu vertailudatan epäkelpoisuus voi hyvin johtua esimerkiksi siitä, että prosessi ei ole stabiili, vaan siinä on kokoajan jonkin asteista värinää. Tämä saattaa osittain vaikuttaa tulokseen ja mallin oikeellisuuteen.

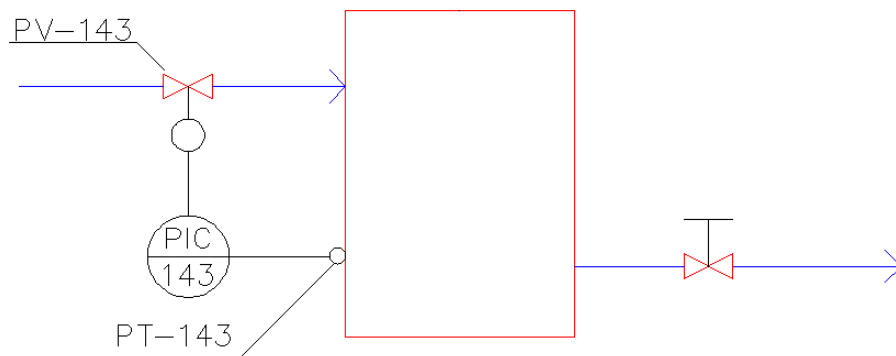


Kuva 51. Painesäiliön trendikäyrä hyödyntäen isompia kertoimia

### 6.2.3 Painesäiliö

Tein askelvastekokeita Kymenlaakson ammattikorkeakoulun automaatiolaboratoriossa. Harjoitusprosessina toimi painesäiliö, jonka painetta säädetään säätöventtiilillä. Painesäiliön alaosassa on putki ja käsikäyttöinen venttiili, johon voimme yhdistää esimerkiksi paineilmaa tarvitsevan pneumaattisen työkalun. Koetta tehdessäni käyttökohdetta ollut, vaan venttiili oli hieman auki ja sitä peitti talouspaperi, jolla minimoitiin äänihaitat.

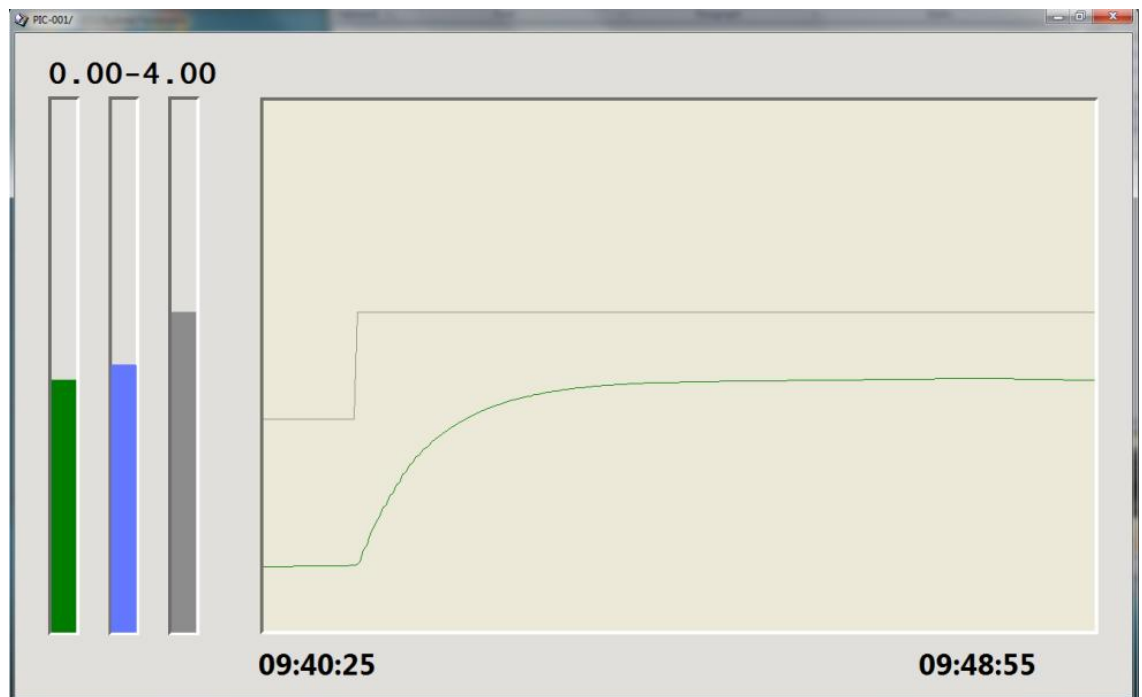
Aloitin työni tutkimalla prosessia ja etsimällä mahdollisia epätarkkuutta aiheuttavia tekijöitä ja mahdollisia vaikeuksia, joita tulisin askelvastekokeen aikana kokemaan. Kymenlaakson ammattikorkeakoulussa on käytössä Metso DNA automaatiojärjestelmä, jonka kautta ohjasin prosessia. Ensimmäinen epätarkkuutta aiheuttava tekijä oli trendikäyrän epätarkkuus. Trendikäyrää sai otettua vai todella suurissa paloissa (7.5 min, 1 h ja 2 h) ongelmaa olisi lievittänyt hyvä Zoom-toiminto tai skaalaus. Kaikkein paras vaihtoehto olisi ollut, että trendikäyrää olisi saanut jälkeensä talteen. Tällaisella toiminnolla voisi esimerkiksi kelailla trendiä, zoomata ja verrata eri kohtia. Aika täytyi arvioida liikkuvasta trendikäyrästä, mikä ei ollut helppoa, koska prosessissa oli viivettä todella vähän. Toinen epätarkkuutta aiheuttava tekijä oli painetasen vaihtelut. Paineen ollessa alhaalla, se kasvoi itsekseen, vaikka säätöventtiiliä ei ohjattu. Tämä saattoi johtua säätöventtiilin vuotamisesta. Paineen ollessa yli yhden barin, paineen kasvu oli melko minimaalista, joten päätin poistaa turhat desimaalit pois.



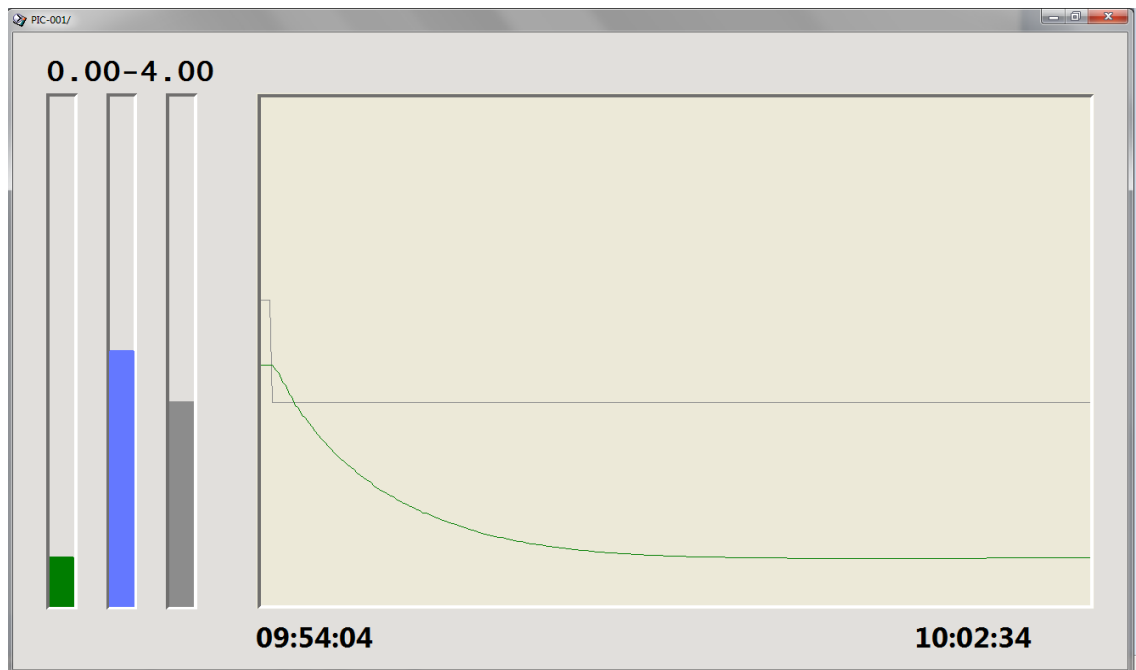
Kuva 52. Painesäiliön PI-kaavio

Päätin tehdä kokeen siten, että teen ensimmäiseksi askelvasteen ylöspäin (kuva 53) ja toisen alaspäin (kuva 54) ja näin varmistua, että sama dynamiikka pätee kumpaankin suuntaan. Aikaisemmin mainitun paineen nousun takia odotin, että paine oli itseksensä kasvanut 0,5 bariin. Tällöin ohjausarvo oli 40 % ja nostin ohjausarvon 60 %:iin. Kuvassa näkyy, että kyseessä on selvästi ensimmäisen kertaluvun ja viiveen prosessi. Tosin viivettä on todella vähän. Kirjoitin askeleen antamisajan ja prosessin reagointiajan ylös ja jäin odottelemaan, että prosessi tasaantuu. Prosessin tasaantumisen tulkitseminen oli painevaihtelun takia hieman haastavahkoa, vertasin paineen vaihtelua ja katsoin, millä kohtaa on eniten tasaista ja otin tästä kohtaa ajat sekä paineen arvot ylös. Paineen vaihtelut osoittautuivat merkityksettömiksi.

Annoin prosessin tasaantua ja laitoin ohjaukselle arvon takaisin 40 %. Otin jälleen paineen, askeleen aloitusajan ja prosessin viiveajan ylös. Kuvista 55 ja 56 voimme huomata, että sama dynamiikka pätee kumpaankin suuntaan ja tulokset ovat lähestulkoon verrannolliset. Kuvissa 53 ja 54 harmaa viiva on prosessiin tehty ohjauksen muutos ja vihreä viiva on prosessin paineen muutos.

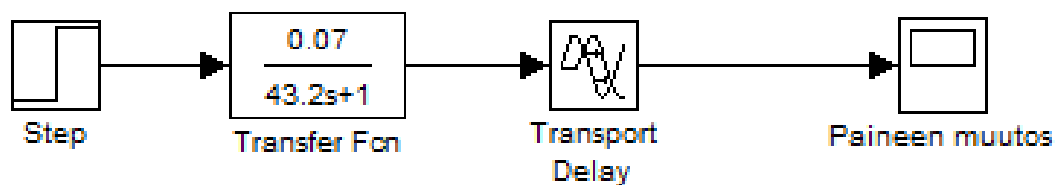


Kuva 53. Askelkoe tehtynä ylöspäin



Kuva 54. Askelkoe tehtynä alaspäin.

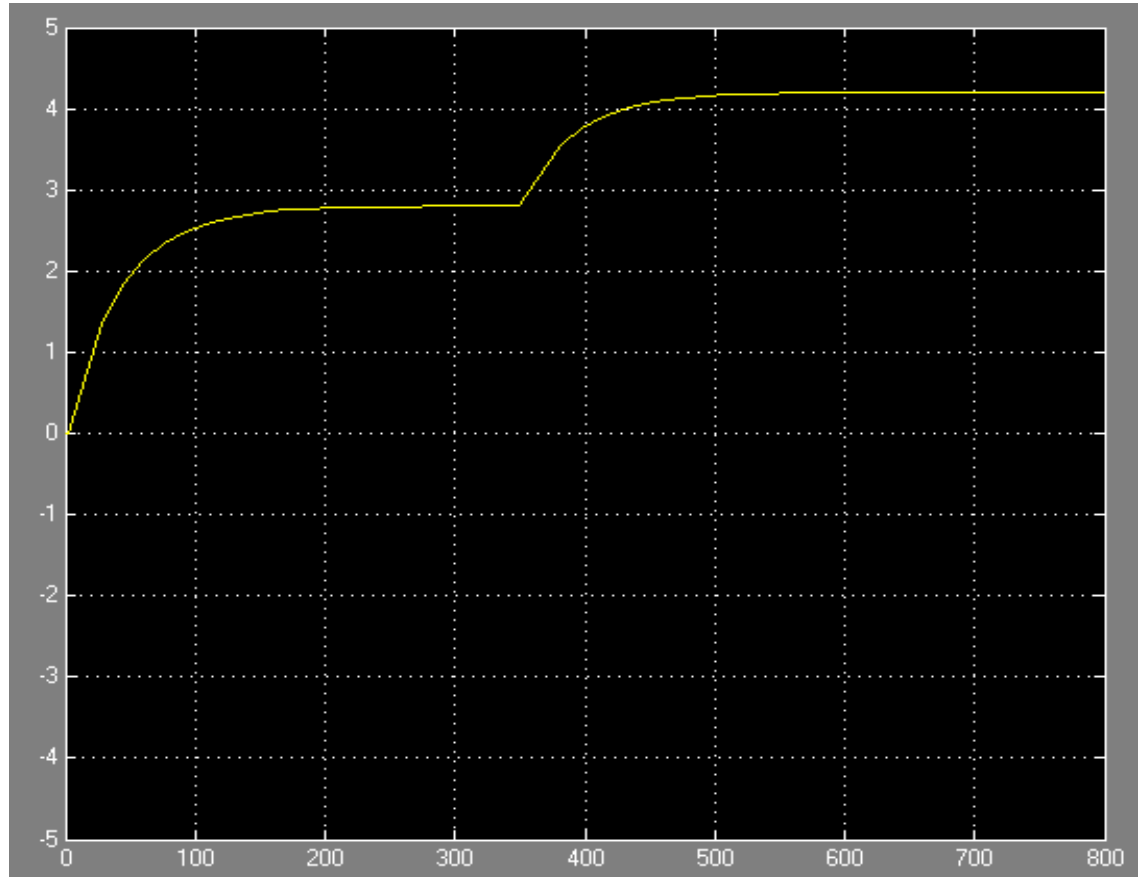
Askelkokeesta saaduilla tuloksilla pystyin toteamaan ensimmäiseksi, että kyseessä on ensimmäisen kertaluvun ja viiveen malli, johon pätee aikaisemmin mainittu kaava 9. Kirjoitin prosessissa tapahtuneiden muutosten aikavälit ylös ja määrittelin trendikäyrästä saaduilla tiedoilla prosessin vahvistuksen, viiveen ja aikavakion. Prosessin vahvistukseksi sain 0.07, viiveeksi sain 4 sekuntia ja aikavakioksi approksimoin 43.2 sekuntia. Prosessin aikavakio on laskettu paineen muutoksen ja ohjauksen muutoksen suhteella. Aikavakion jouduin arvioimaan paperitulosteen perusteella. Tämän perusteella rakensin Simulink-ohjelmalla lohkokaaviomallin, joka on kuvassa 55.



Kuva 55. Painesäiliön Simulink-ohjelma

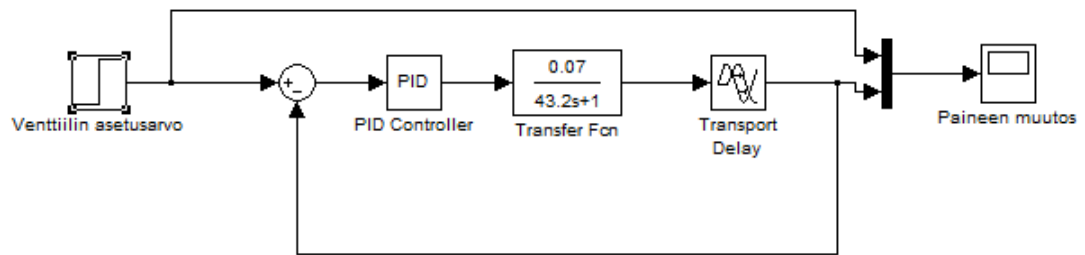
Tein prosessiin vastaavan muutoksen, kuin koulun automaatiolaboratoriossa ja sain kuvan 56 mukaisen trendikäyrän. Simulink aloittaa prosessin 0 lukemasta, joten päätin odottaa, että prosessi tasaantuu ja laitoin prosessin tekemään askelmaisen muutoksen

350 sekunnin kohdalla. Tuloksena sain, että paine nousee n. 0.45 baria. Tämä ei vastaa saamiamme mittaustuloksia jossa 20 % muutoksesta tuli 1.4 barin muutos. Kokeilujen perusteella ongelma koskee K-termiä eli 0.07. Yritin laskea uudelleen, mutta en löytänyt virhettä laskuista, enkä kaavoista, joten virhe on voinut tulla prosessista.

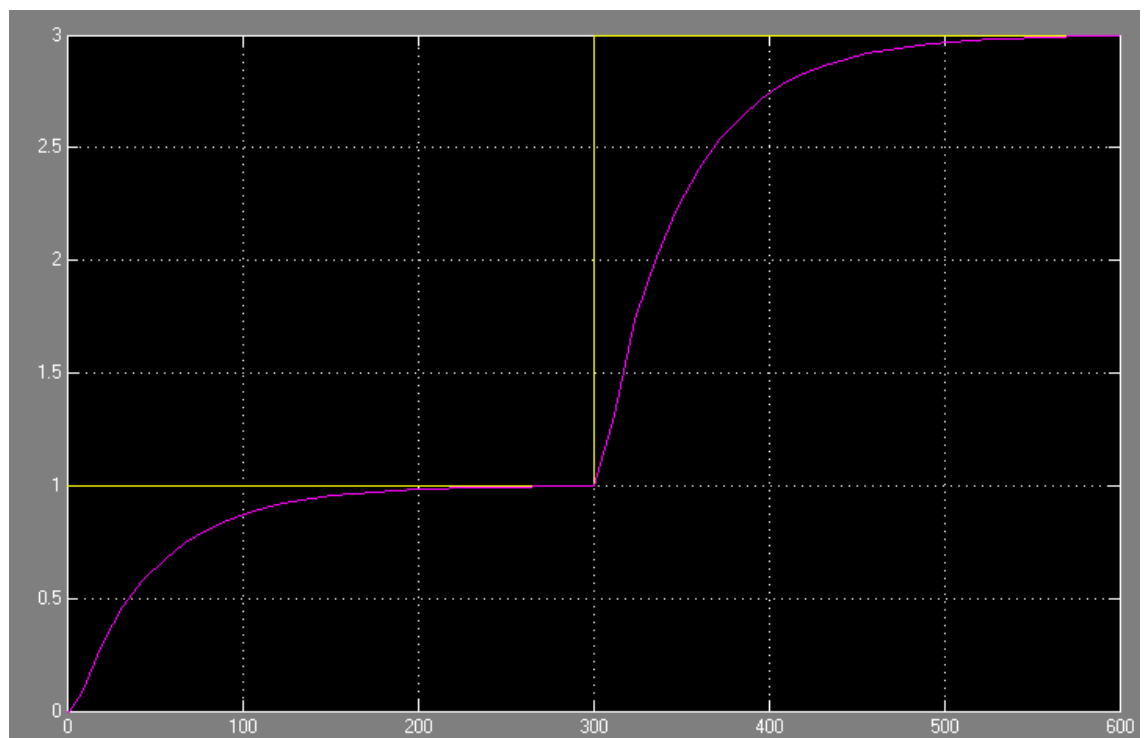


Kuva 56. Painesäiliön askelvastekoe simuloituna

Laadin prosessista myös mallin, jossa on mukana säädin. Säätimen lisääminen vaatii summaimen ja PID-lohkon, jossa vahvistus (K) ja integrointiajan (I) arvot laitetaan hieman eri tavalla, kuin normaalisti (kuva 57). K-arvo laitetaan normaalisti, mutta I-arvoksi laitetaan K-arvon ja I-arvon osamäärä. Laskin lambda-virityksen avulla ( $\lambda$ -arvo 1.1) parametreiksi, että vahvistus on 11.98 ja integrointiaika on 0.277. Nämä arvot ovat laskettuna valmiiksi Matlabin säätimen haluamaksi parametriksi. Tein tällä kertaa simuloinnin muuttamalla asetusarvoa 2 baria, kun 350 sekuntia oli kulunut. Simuloinnin tuloksena saatiin kuvan 58 mukainen trendikäyrä, jossa keltainen viiva on venttiilin asetusarvon muutos ja violetti viiva paineen muutos. Voimme todeta trendikäyrästä, että säätö toimii hitaasti, mutta vakaasti ja varmasti.



Kuva 57. Painesäiliö prosessin Simulink-ohjelma PID-säätimellä varustettuna



Kuva 58. Asetusarvon muutoksesta saatu tulos

Sain tietää askelkokeiden ja simuloinnin jälkeen, että venttiili toimii epälineaarisesti. Tämä tarkoittaa sitä, että prosessi toimii optimaalisesti vain tietyllä välillä. Tämä selittää osittain simuloinnissa nähdyt epätarkkuudet. Tuloksista kuitenkin voidaan todeta, että askelkokeella saadaan helposti yksinkertainen malliprosessista. Lisäksi prosessin tulokset on helppo laittaa Simulink-ohjelmistoon ja testata uudelleen, vastaako malli todellisuutta.

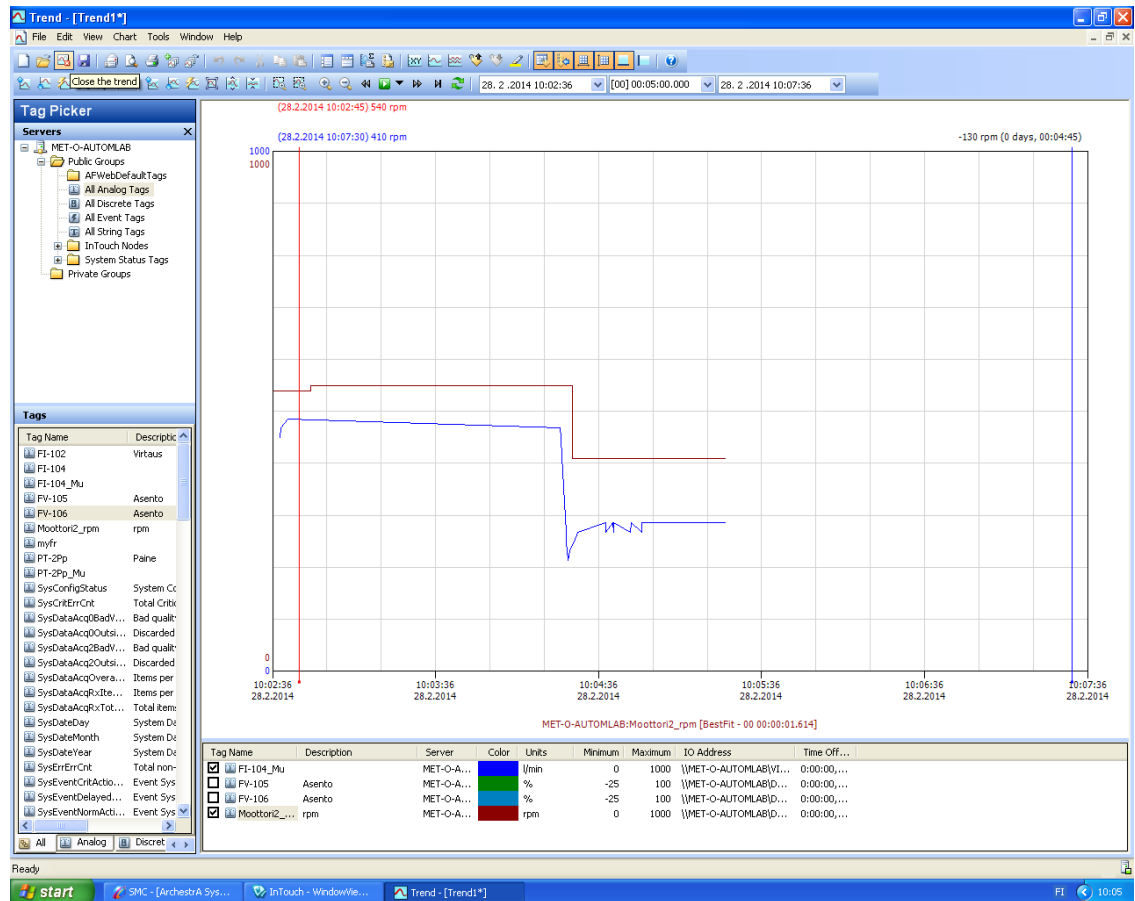
#### 6.2.4 Prosessin tiedonkeruussa tapahtuvat ongelmat

Prosessin mallintamisessa tiedon kerääminen ei aina onnistu niin kuin pitäisi. Tarkoituksenani oli tehdä opinnäytetyöhön esimerkki, jossa keräisin tietoa koulussa olevasta vesiprosessista. En alussa tiennyt prosessista mitään, joten asiat tulivat minulle yllätyksenä. Ennen prosessikokeita, prosessi kannattaa tutkia perin pohjin. Lisäksi myös kaikki tiedonsiirtoon liittyvät asiat kannattaa tarkistaa.

Alussa kokeilin tehdä askelvastekokeen, jossa ohjaisin säätöventtiiliä. Säätöventtiilin avaaminen nostaisi virtauksen määrää. Prosessin ollessa päällä, säädin venttiiliä avautumaan n. 20 %, mutta prosessi reagoi hyvin erikoislaatuksella tavalla. Venttiilin ohjauskäyrä, jonka piti olla askel, liikkui ikään kuin portaittain. Porrasmainen askel ei vastaa prosessin todellista kuvaa.

Toisella kerralla saimme selvitettyä, kuinka prosessista saa kerättyä dataa Excel-tilukoksi. Vaihdoin säätöventtiilin pumppuun ja tarkoituksena oli muokata virtausnopeutta askelmaisesti. Tässä tapauksessa epätarkkuutta aiheutti, että pumppun nopeutta säädettiin ohjauspulpetissa olevalla potentiometrillä. Päätin tehdä askeleita välillä 2-3. Aloitin prosessin ylösajon ja annoin sen tasaantua. Tässä kohtaa huomasi, että prosessissa esiintyy outoja kuormitushäiriöitä. Prosessi tasaantui ja tein askelmaisesta muutoksen, jolloin trendikäyrästä näkyi, että prosessi muuttui nopeammin kuin askel oli tullut. Tein vastaavanlaisia askeleita useampia, mutta aina tapahtui sama asia. Otin tiedot talteen, tosin tässäkin kohtaa tuli ongelma, kun tietokone ei hyväksynyt jostain syystä USB-muistitikkuani. Saamani Excel-tilukon perusteella ei datasta voitu tehdä ARX-mallia, koska tiedot eivät päivittyneet kunnolla. Aina silloin, kun mootorin nopeus oli päivitetty, virtausnopeutta ei ollut päivitetty. ARX-mallia varten olisi tarvittu tietoa säännöllisin väliajoin ja siten, että tiedot olisi otettu samaan aikaan.





Kuva 59. Epäonnistunut askelkoe

Kysyin asiasta Kymenlaakson Ammattikorkeakoulun lehtorilta Vesa Kankkuselta asiasta ja sain tietää, että pumpun nopeustiedossa on maksimissaan 6 sekunnin viive. Tämä tiedon pohjalta on vaikea rakentaa mallia, koska tiedot ovat melko epätarkkoja. Lisäksi prosessi käyttäytyi hyvin epälineaarisesti. Kappaleessa 6.2.2 mainitsin, että painesäiliön säätöventtiili, jota ohjasin, oli epälineaarinen. Epälineaarisuutta ei ollut havaittavissa tehdessäni askelkoetta. Tässä prosessissa huomasin epälineaarisuuden ja epästabiilisuuden, koska prosessi ei käyttäytynyt tasaisesti niin kuin odotin.

### 6.3 Luonnonlakimalli vai kokeellinen malli?

Luonnonlakimallilla ja kerättyyn tietoon perustuvalla mallilla ei ole välttämättä valtavia eroja, vaan niiden käyttäminen riippuu siitä, millainen prosessin tapaus on. Esimerkiksi jos prosessissa on tiedonkeruu mahdollisuus, niin se kannattaa hyödyntää, koska se on kyseisessä tapauksessa paljon helpompaa. Mikäli prosessi on suunnittelusteella, kannattaa käyttää luonnonlakimallia, jolloin voidaan testata mitoituksen toimivuutta.

Mallien tarkkuus riippuu hyvin monesta tekijästä. Luonnonlakimalleissa pitää huomioida suureiden tarkkuus, ettei mennä ylitarkoiksi, esimerkiksi pinta-alojen kanssa. Kokeellisessa mallintamisessa tarkkuus tulee antureiden antamista tiedoista ja siitä, kuinka paljon tietoa on saatavilla. Tässäkin tapauksessa tulee muistaa, että dataa ei saa olla liian vähän, eikä liian paljon.

Kumpikin mallintamistapa on hyvä ja ne tukevat toisiaan. Esimerkiksi suunnitellun prosessin mallin vastaavuus voidaan rakennusvaiheen jälkeen testata ja katsoa, onko prosessista saadut suureet juuri sellaiset, kuin on haluttu ja, jos prosessissa ongelmia niin millaisia.

## 7 YHTEENVETO

Prosessiteollisuuden prosessien hallitseminen vaatii prosessin tuntemista ja prosessin dynamiikan tuntemista muutostilanteissa. Matemaattinen mallinnus ja simulointi ovat hyödyllisiä ja käyttökelpoisia. Niiden avulla voimme ymmärtää hallittavaa prosessia paremmin. Matemaattinen mallintaminen vaatii runsaasti resursseja, mutta sen avulla voidaan saada parempi tuottavuus ja turvallisuus. Lisää tuottavuutta syntyy, kun prosessi optimoidaan toimimaan mallista saadun tiedon avulla. Esimerkiksi säätimen viittäminen askelkokeen perusteella on yksi tapa optimoida prosessinohjausta.

Mallirakenne määräytyy sen mukaan, onko prosessi olemassa vai vasta suunnitteluasteella. Tärkeimmät dynaamiset mallirakenteet ovat: ensimmäisen kertaluvun ja viiveen malli, integroivamalli ja ARX-malli. Nämä mallirakenteet luovat vahvan pohjan simulointia varten. Mallirakenteeksi löytyy myös muitakin eri vaihtoehtoja, kuten summa logiikka ja neuroverkot, mutta niiden käyttöön ottaminen vaatii huomattavasti enemmän resursseja.

Opinnäytetyön aihe oli kiinnostava ja se antoi minulle uusia näkemyksiä prosessiteknikasta. Aihe oli minulle tuttu aikaisemmista kursseista ja tiesin sen olevan hyvin laaja-alainen, joten sen rajaaminen oli tärkeää. Työn tekeminen oli mieltä, koska aihealue oli minulle tuttu aikaisemmin suoritetuista kursseista. Opinnäytetyöstä on myös apua mahdollisia jatko-opintoja ajatellen.

## LÄHTEET

**Painettumateriaali:**

Dorf, R.C. Bishop, R.H. 2011. Modern Control Systems Twelfth Edition. Upper Saddle River (New Jersey): Pearson.

Glanville, Dr. A. R. 2013. Scientifica. 1. p. Australia, Millenium House Pty Ltd.

Hansen, N. 2013. Sääkartta täyttää 150 vuotta. Tieteen kuvalehti 11/2013, s.64 - 69.

Harju, T. Marttinen, A. 2000. Säättöpiirinvirityksen perusteet. 1. painos. Espoo: Control CAD Oy.

Jacobien, R. K 2013. Neljästä hankkeesta syntyvät keinotekoiset aivot. Tieteen kuvalehti 11/2013, s. 18 - 23.

Jäppinen, P. Kupiainen, A. Räsänen, M. 2005. Lukion calculus: 1 : Funktiot ja yhtälöt; Polynomifunktiot. 1-2. painos. Helsinki: Otava.

Joronen, T. Kovács, J Majanne Y. 2007. Voimalaitosautomaatio. 2. painos. Helsinki: Suomen automaatioseura.

Leiviskä, K. 1999. Papermaking science and technology: Book 14: Process control. Helsinki: Fapet Oy.

Marlin, T. E.1995. Process Control: Designing Processes and Control Systems for Dynamic Performance. New York: McGraw-Hill, Inc.

Råde, L. Westergen, B. 2004.Mathematics Handbook for Science and Engineering. Sweden: Studentlitteratur AB.

Savolainen, J. Vaittinen, R. 1995. Säättötekniikan perusteita. Helsinki: Opetushallitus.

Tekniikan kaavasto. 2005. 5. uudistettu painos. Tampere: AMK-Kustannus Oy.

**Internetlähteet:**

About Scilab Enterprises. s.a. Scilab Enterprises:n internetsivut. Saatavissa:  
<http://www.scilab-enterprises.com/company/aboutus> [viitattu 24.3.2014].

Boyd, S. s.a. The Laplace transform. Internetsivut. Saatavissa:  
<http://www.stanford.edu/~boyd/ee102/laplace.pdf> [viitattu 12.7.2013].

Company Overview. s.a. Matlabin internetsivut. Saatavissa:  
<http://www.mathworks.se/company/aboutus/index.html> [viitattu 24.3.2014].

Differential Equations. s.a. Hyper Physic:n internetsivut. Saatavissa:  
<http://hyperphysics.phy-astr.gsu.edu/hbase/diff.html> [viitattu 16.9.2013].

First Order Non-Homogenous differential equation. s.a. Hyper Physic:n internetsivut.  
 Saatavissa: <http://hyperphysics.phy-astr.gsu.edu/hbase/math/deinhom.html#c1> [viitattu 16.10.2013].

Haapanen, E. 2004. Tuulivoimatekniikka. PowerPoint-esitys 20.05.2004. Tuulitaidon internetsivut. Saatavissa:  
[http://www.tuulitaito.fi/Artikkelit/tuulivoimatekniikkaa\\_luento.ppt](http://www.tuulitaito.fi/Artikkelit/tuulivoimatekniikkaa_luento.ppt) [viitattu 18.5.2014].

Harju, T. Marttinen, A. 2000. Sääätötekniikan koulutusmateriaali.internetsivu. Saatavissa:  
<http://www.automaatioseura.com/automaatioseura/tiedostot/finish/17/155> [viitattu 27.6.2013].

Haugen, F. 2009. Padé-approximation of a time-delay. internetsivu. Saatavissa:  
[http://home.hit.no/~hansha/documents/control/theory/pade\\_approximation.pdf](http://home.hit.no/~hansha/documents/control/theory/pade_approximation.pdf) [viitattu 11.7.2013].

Hietanen, T. 1997. Viiveellisen prosessin säätö. Oulun ammattikorkeakoulun internetsivu. Saatavissa: [http://www.tekniikka.oamk.fi/~tero/Labrat/C\\_prediktori.doc](http://www.tekniikka.oamk.fi/~tero/Labrat/C_prediktori.doc) [viitattu 10.3.2014].

Honkela, T. 1998. Neuroverkot: johdatus moderniin tekoälyyn. Aalto-yliopiston internetsivut. Saatavissa: <http://users.ics.aalto.fi/tho/stes/step96/honkela2/> [viitattu 20.5.2014].

Kotilainen, S. 2013. Eksoottinen kvanttietokone tulee arkikäyttöön. mBnet-lehden internetsivut. Saatavissa: [http://www.mbnet.fi/artikkeli/ajankohtaiset/ajassa/eksoottinen\\_kvanttietokone\\_tulee\\_arkikayttoon](http://www.mbnet.fi/artikkeli/ajankohtaiset/ajassa/eksoottinen_kvanttietokone_tulee_arkikayttoon) [viitattu 7.6.2013].

Kotilainen, S. 2013. Kiina superkoneiden valtiaaksi hurjalla erolla. Tietokone-lehden internetsivut. Päivitetty 7.6.2013. Saatavissa: [http://www.tietokone.fi/artikkeli/uutiset/kiina\\_superkoneiden\\_valtiaaksi\\_hurjalla\\_erolla](http://www.tietokone.fi/artikkeli/uutiset/kiina_superkoneiden_valtiaaksi_hurjalla_erolla) [viitattu 7.6.2013].

Mathcad opiskelijaversio. s.a. Mathcadin internetsivut. Saatavissa: <http://mathcad.fi/opiskelu.html> [viitattu 23.5.2014].

Vänskä, O. 2012. Kiinnostaako sadan tonnin palkkio? Todista kvanttietokone mahdolltomaksi. MPC-lehden internetsivut. 7.2.2012. Saatavissa: [http://www.mikropc.net/kaikki\\_uutiset/kiinnostaako+sadan+tonnin+palkkio+todista+kvanttietokone+mahdolltomaksi/a772078](http://www.mikropc.net/kaikki_uutiset/kiinnostaako+sadan+tonnin+palkkio+todista+kvanttietokone+mahdolltomaksi/a772078) [viitattu 24.7.2013].

PID-säädön perusteet. 2011. Oulun yliopiston internetsivut. Saatavissa: [http://cc.oulu.fi/~ylikoant/PID/PID\\_saadon\\_perusteet\\_2011.pdf](http://cc.oulu.fi/~ylikoant/PID/PID_saadon_perusteet_2011.pdf) [viitattu 21.6.2013].

Products for MATLAB & Simulink Version R2014a. s.a. Matlab:n internetsivut. Saatavissa: [https://www.mathworks.se/store/link/products/standard/new?s\\_iid=htb\\_buy\\_gtwy\\_cta4](https://www.mathworks.se/store/link/products/standard/new?s_iid=htb_buy_gtwy_cta4) [viitattu 18.4.2014].

Products for MATLAB Home R2014a. s.a. Matlab:n internetsivut. Saatavissa: [https://www.mathworks.se/store/link/products/home/new?s\\_iid=htb\\_buy\\_gtwy\\_cta4](https://www.mathworks.se/store/link/products/home/new?s_iid=htb_buy_gtwy_cta4) [viitattu 18.4.2014].

SCILAB License. 2014. Scilab Atoms Portal -internetsivut. Saatavissa: <https://atoms.scilab.org/licenses/17> [viitattu 24.5.2014].

Simulink Product Description. s.a.Matlabin internetsivut. Saatavissa:  
<http://www.mathworks.se/help/simulink/gs/product-description.html> [viitattu 24.3.2014].

Simulointi. s.a. Ramentorin:n internetsivut. Saatavissa:  
<http://www.ramentor.com/etusivu/teoria/simulointi/> [viitattu 28.8.2013].

Smith III, J.O. 2007. Difference Equation. Julius Orion Smith III:en internet sivut. Saatavissa: [https://ccrma.stanford.edu/~jos/filters/Difference\\_Equation\\_I.html](https://ccrma.stanford.edu/~jos/filters/Difference_Equation_I.html) [viitattu 16.7.2013].

System Requirements – Release R2014a. s.a.Matlabin internetsivut. Saatavissa:  
[http://www.mathworks.se/support/sysreq/current\\_release/](http://www.mathworks.se/support/sysreq/current_release/) [viitattu 24.3.2014].

Säätö- ja ohjaustekniikka. 2005. internetsivut. Saatavissa:  
[http://personal.inet.fi/yritys/kkov.eduserver/yhteinen/saato\\_&\\_ohj\\_tekniikka25\\_45.pdf](http://personal.inet.fi/yritys/kkov.eduserver/yhteinen/saato_&_ohj_tekniikka25_45.pdf) [viitattu 18.2.2014].

Tilaa Mathcad. s.a. Mathcad:n internetsivut. Saatavissa: <http://mathcad.fi/tilaa.html> [viitattu 16.10.2013].

Tham, M.T. 2000. Dynamic Models For Controller Design. Newcastleen yliopiston verkkosivut. Saatavissa: [lorien.ncl.ac.uk/ming/digicont/mbpc/models.pdf](http://lorien.ncl.ac.uk/ming/digicont/mbpc/models.pdf) [viitattu 18.5.2014].

### **Luentosarjat:**

Lyytikäinen, T. Mäkelä, M. 2014. Kehittyneet säätömenetelmät. Luentosarja. Kevät 2014. Kotka: Kymenlaakson ammattikorkeakoulu.

Mäkelä, M. 2013. Automaatiosuunnittelu. Luentosarja. Syksy 2013. Kotka: Kymenlaakson ammattikorkeakoulu.

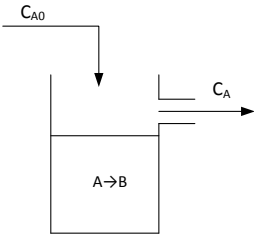
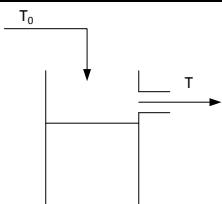
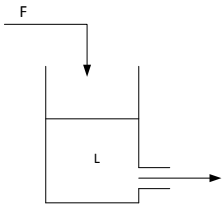
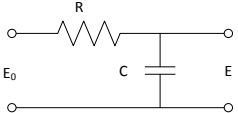
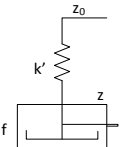
Mäkelä, M. 2013. Kenttäväylät. Luentosarja. Kevät 2013. Kotka: Kymenlaakson ammattikorkeakoulu.

Staben, V. 2013. Modeling and Simulation – Introduction and Examples. Luentosarja. Kevät 2013. Kotka: Kymenlaakson ammattikorkeakoulu.

Ristola, E. 2013. Prosessidynamiikka. Luentosarja. Kevät 2013. Kotka: Kymenlaakson ammattikorkeakoulu.

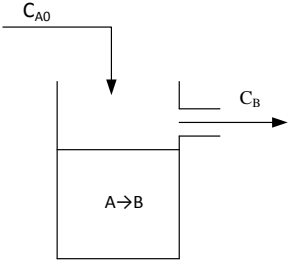
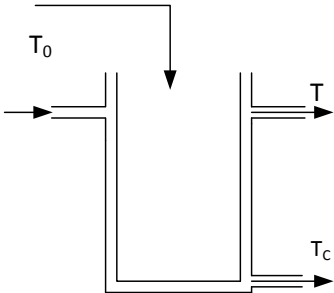
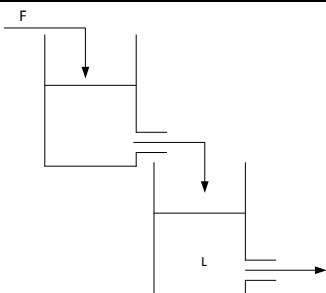
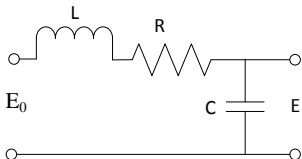
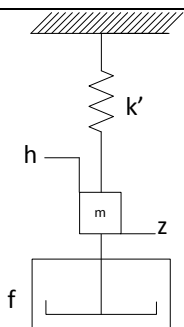


## LIITE 1.

Prosessi	Tase	Tulo	Lähtö	$K_p$	$\tau$
	kemiallinen reaktio	$C_{A0}$	$C_A$	$\frac{F}{F + V_k}$	$\frac{V}{F + V_k}$
	energia	$T_0$	$T$	1.0	$\frac{V}{F}$
	kokonais- massa	$F$	$L$	$\frac{1}{0.5kL_s^{-0.5}}$	$\frac{A}{0.5kL_s^{-0.5}}$
	virta	$E_0$	$E$	1.0	$R \cdot C$
	voima	$z_0$	$z$	1.0	$\frac{f}{k'}$

$C$	konsentraatio
$T_0$	tuotulämpötila
$T$	lämpötila
$E_0$	tulojännite
$E$	jännite
$z_0$	siirtymäalussa (oletettavasti 0)
$z$	kokonaissiirtymä
$f$	kitkakerroin
$k$	reaktionopeusvakio (määritellään kokeellisesti)
$k'$	jousivakio (Marlin 1995, 151)

## LIITE 2.

Systeemi	Tase	Tulo	Lähtö	$K_p$	$\tau^2$	$2\xi\tau$
	kemiallinen reaktio	$C_{A0}$	$C_B$	$\frac{Vk}{F + Vk}$	$\frac{V}{F + Vk}$	$\tau_A + \tau_B$
	energia	$T_c$	$T$	1.0	$\frac{V}{F}$	
	kokonaismassa	$F$	$L$	$\frac{1}{0.5kL_s}$	$\frac{A}{0.5kL_s^{-0.5}}$	$2\tau$
	virta	$E_0$	$E$	1.0	$RC$	$RC$
	voima	$h$	$z$	$\frac{1}{k'}$	$\frac{m}{k'}$	$\frac{f}{k'}$

$\xi$	höyristymisvakio
$E$	jännite
$z$	sijainti
$k'$	jousivakio
$f$	värähtelyvakio
$h$	voima
$m$	massa
$T_a$	$(V/F + V_k)$
$T_b$	$(V/F)$ (Marlin 1995, 152)